



Office de la propriété  
intellectuelle  
du Canada

Un organisme  
d'Industrie Canada

Canadian  
Intellectual Property  
Office

An Agency of  
Industry Canada

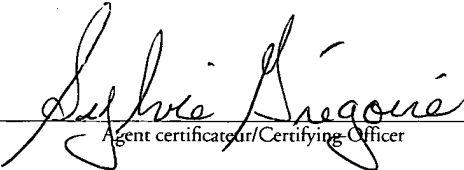
*Bureau canadien  
des brevets  
Certification*

La présente atteste que les documents  
ci-joints, dont la liste figure ci-dessous,  
sont des copies authentiques des docu-  
ments déposés au Bureau des brevets.

*Canadian Patent  
Office  
Certification*

This is to certify that the documents  
attached hereto and identified below are  
true copies of the documents on file in  
the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No:  
**2,414,378**, on December 9, 2002, by **COREL CORPORATION**, assignee of Gordon  
Bowman and Peter Barrett, for "System and Method for Controlling User Interface of a  
Web Application".

  
Agent certificateur/Certifying Officer

July 16, 2003

Date

Canada

(CIPO 68)  
04-09-02

OPIC  CIPO

Abstract

A system and method of controlling user interface features of a web application is provided. The system comprises a collection of user interface control elements, a  
5 collection of skin templates comprising extensible markup language based markup contained as children of a container element, a collection of control element scripts for performing actions associated with the control elements, and an initialization function for directing the processing of one or more control elements in a document object model. Each control element comprises a name following a predetermined naming convention,  
10 common attributes for defining graphical features of the control element and for associating the control element with the internal state of the core control element, other attributes for defining attributes that affect the intrinsic behavior of the control, and a skin template reference attribute for referencing a skin template. Each script associated with a control element.

## **System and Method for Controlling User Interface Features of a Web Application**

### **FIELD OF THE INVENTION**

The invention relates to markup languages. In particular, the invention relates to a  
5 system and method for controlling user interface features of web applications.

### **BACKGROUND OF THE INVENTION**

Web designers use markup languages to create and modify web sites. Graphical  
User Interfaces (GUI's), also known as user interface (UI) controls, are desirable for  
10 building web applications, as they allow the user to interact with the application in a  
standard, familiar way. UI controls are high-level constructs with data and function  
methods underlying a visual front-end. They behave in standard, expected  
manners—changing their states (e.g., selected, unselected, focussed, disabled, etc.), which  
usually have different appearances, in response to user interaction. For example, clicking  
15 on a push-button will cause the button to go into its down-state in response to the  
'mousedown' event and back into its up-state in response to the 'mouseup' event.  
Another intrinsic behavior of UI controls is that they trigger the execution of  
predetermined actions in response to specified events, such as the user clicking on them  
with the mouse.

20 Hypertext Markup Language (HTML) is currently the most commonly used  
markup language, or platform, with which to build web applications. HTML is limiting,  
with poor graphics support (only bitmaps, no vectors), its lack of absolute positioning, its  
small set of user interface (UI) controls and its inability to describe the appearance of  
those UI controls (the HTML browser decides on the appearance, and it cannot be  
25 overridden). Scalable vector graphics (SVG) is a language for describing two-  
dimensional graphics in extensible markup language (XML), which may soon become the  
platform of choice with which to build web applications. SVG provides an author with a  
rich set of vector graphics features (as well as bitmaps) and absolute control over their  
positioning.

30 SVG-based web applications create their own UI controls using SVG to describe  
the visual appearance and script to describe the data and function members, which  
provide the UI controls with the standard intrinsic behaviors discussed previously. This

means, however, that non-developers cannot author the UI portion of the content, a fair amount of script for the UI controls must be transferred to the client (a significant problem for wireless devices) and the UI controls may seem to react sluggishly to user input, due to the nature of script being interpreted at run-time as opposed to native code (e.g. C or C++) being interpreted (and optimized) at compile time.

XUL is an XML markup language for UI controls created by the Mozilla group, specifically for use in their Mozilla HTML web browser. While XUL does offer an extensive list of UI controls, offers limited skinnability (control of appearance) and is not tied to a forms creation/processing/submission model, XUL is intended for use in HTML browsers and thus does not allow for absolute positioning nor absolute control of the appearance, cannot be easily hooked up to actions and cannot be easily used for forms. XUL is not supported in the Adobe (TM) SVG Viewer.

XForms is another XML markup language which includes UI controls. XForms was designed for forms, however. XForms only supports UI controls needed for forms, does not allow for absolute control over their positioning and their appearance, cannot be easily hooked up to actions (except those that affect just its own UI controls and its "instance", or forms, data) and is very tightly tied to its forms model. XForms is not supported in the Adobe SVG Viewer.

Figure 1 shows a typical web display environment 10 for displaying web pages and web applications. A web display environment 10 comprises a browser 11, a viewer 13, a script interpreter 14, and a DOM 15. The browser 11 is the host application, which understands and visually renders hypertext markup language (HTML) and/or extensible hypertext markup language (XHTML). Examples of browsers include Netscape (TM) and Internet Explorer (TM). The browser 11 includes a window which is displayed on the display apparatus, such as a monitor, of an end user computer system. The browser 11 typically employs a plug-in architecture, in which third party software (known as the plug-in or viewer 13) can be associated with any file format that is not already natively supported by the browser 11 and is allowed to render that file within the host browser's 11 window. One type of file that the browser 11 may be asked to open is a Scalable Vector Graphic (SVG) file having a ".svg" extension. The browser 11 does not natively support the SVG markup language (which is an XML language) and so passes the SVG

file to the SVG viewer 13, which has associated itself to the SVG file format, via the rules of the plug-in architecture of the browser 11.

The viewer 13 comprises software code for parsing the SVG markup, creating a DOM, rendering that DOM to the browser's window, listening for events and dispatching them to their assigned handler script functions, and interpreting/executing those script functions. An example of a viewer 13 is the Corel (TM) SVG Viewer. The viewer 13 uses the SVG file received from the browser 11 to create a DOM 15. The DOM is a hierarchical tree structure of objects in memory, representing the hierarchical XML markup in the XML text file. The DOM also contains methods (also known as functions or application programming interfaces (API's)) that allow it to be queried or modified. The viewer 13 may also have access to a script interpreter/engine, which can execute script code 14 created by a programmer for the purpose of making the document non-static (e.g., animation) and/or interactive with the user (e.g., the user can create events with the mouse or keyboard, which cause something to happen) via manipulation of the DOM.

#### SUMMARY OF THE INVENTION

It is an object of the invention to provide a novel system and method of controlling user interface features of a web application that obviates or mitigates at least one of the problems described above.

In an aspect of the present invention, there is provided a system for controlling user interface features of a web application. The system comprises a collection of user interface control elements, a collection of skin templates comprising extensible markup language based markup contained as children of a container element, a collection of control element scripts for performing actions associated with the control elements, and an initialization function for directing the processing of one or more control elements in a document object model. Each control element comprises a name following a predetermined naming convention, common attributes for defining graphical features of the control element and for associating the control element with the internal state of the core control element, other attributes for defining attributes that affect the intrinsic

behavior of the control, and a skin template reference attribute for referencing a skin template. Each script associated with a control element.

In another aspect of the present invention, there is provided a system for controlling user interface features of a web application. The system comprises a  
5 collection of control element scripts for performing actions associated with the control elements, and an initialization function for directing the processing of one or more control elements in a document object model. Each script associated with a control element.

In another aspect of the present invention, there is provided a method of controlling user interface features of a web application. The method comprises the steps  
10 of searching for a designated user interface control element in a document object model, and calling a script associated with the designated control element.

In another aspect of the present invention, there is provided a method of controlling user interface features of a web application. The method comprises the steps of adding a behavior element as a child of a user interface control element, receiving an  
15 event which is equal to an event attribute setting in the behavior element, and calling a script associated with the behavior element.

In another aspect of the present invention, there is provided a method of creating a customizable user interface control element having expected behaviours. The method comprises the steps of categorising user interface controls into fundamental core controls,  
20 determining variations of a core controls, determining common attributes of the core control, determining fundamental states for the core control, determining how to allow for absolute positioning of objects the core control, determining how to allow for absolute customization of appearance of the core control, assigning a reference link to the core control, determining templates for skins to allow for the absolute customization of  
25 appearance of the core control, determining how to associate behaviors to the core control, and creating a core control element.

In another aspect of the present invention, there is provided computer readable media storing the instructions and/or statements for use in the execution in a computer of a method of controlling user interface features of a web application. The method  
30 comprises the steps of searching for a designated user interface control element in a

document object model, and calling a script associated with the designated control element.

5 In another aspect of the present invention, there is provided electronic signals for use in the execution in a computer of a method of controlling user interface features of a web application. The method comprises the steps of searching for a designated user interface control element in a document object model, and calling a script associated with the designated control element.

10 In another aspect of the present invention, there is provided a computer program product for use in the execution in a computer of a method for controlling user interface features of a web application. The computer program product comprises a collection of control element scripts for performing actions associated with the control elements, and an initialization function for directing the processing of one or more control elements in a document object model. Each script associated with a control element.

## 15 BRIEF DESCRIPTIONS OF THE DRAWINGS

Figure 1 shows a typical web display environment for displaying web pages and web applications.

Figure 2 shows a user interface control system, in accordance with an embodiment of the present invention.

20 Figure 3 is a node representation of a document object model of an example of a file used in the user interface control system, in accordance with an embodiment of the present invention.

25 Figures 4A to 4D are screen shots of an example of the display of a user interaction with an example of a file used in the user interface control system, in accordance with an embodiment of the present invention.

Figure 5 is a flowchart of an example of a method of creating user interface controls of a web application at load time, in accordance with an embodiment of the present invention.

30 Figure 6 is a flowchart of a method of creating user interface controls of a web application in response to an event, in accordance with the an embodiment of the present invention.

Figure 7 shows another example of a method of controlling user interface features of a web application, in accordance with an embodiment of the present invention.

Figure 8 shows an example of a method of creating an element for controlling user interface features of a web application, in accordance with an embodiment of the present invention.

Figure 9 shows an example of a method of creating a plurality of elements for controlling user interface features of a web application, in accordance with an embodiment of the present invention.

## 10 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 2 shows a system for controlling user interface features of a web application 20 in accordance with an embodiment of the present invention. The user interface (UI) control system 20 comprises a collection (or set) of UI control elements 22, a collection of UI control skin templates 27, a collection of UI control script 28, and an initialization function 21. Each UI control element 22 comprises a name 23, common attributes 24, other attributes 25, and a skin template reference attribute 26. The name 23 is what the UI control element 22 is called (e.g., button, combobox, slider, etc.). As these UI control elements 22 are not currently part of the scalable vector graphics (SVG) specification, their name 23 are prefixed with a namespace (e.g., "dsvg:") so that the viewer's extensible markup language (XML) parser allows it to be part of the DOM. The common attributes 24 are the attributes that are common to every UI control element 22, which define the graphical features of the UI control elements 22 (e.g., width="100"), define the UI control's label or associated text (e.g., label="OK") and position, define the initial state of the UI control element 22 (e.g., disabled="true") and override the default appearance of specific states (described in the associated skin) with new appearances described elsewhere (e.g., down="#skinAlternateDown"). The other attributes 25 define the attributes that are specific to each particular UI control element 22, which affect the intrinsic behavior of the UI control (e.g., toggle="true" causes a button control to become a "sticky" button, staying down or up after each click). An example of an event occurs when a user clicks on a UI control element displayed in a window of a browser. The skin



template reference attribute 26 provides a reference to the location of the UI control skin template 27 which the UI control element 22 may access.

The UI control skin templates 27 are associated with the UI control elements 22. Preferably, there is a one-to-one relationship between UI control elements 22 and UI control skin templates 27. A UI control skin template 27 is designed to allow for absolute customization of appearance of the UI control. The skin template 27 is referenced via an 'xlink:href' attribute (or skin template reference attribute) 26 which is a common attribute of the UI control elements 22. The skin template 27 comprises SVG markup (which can reference bitmap images) contained as children of a 'g' element (a group/container element), where there is one group per state (e.g. 'up', 'down', 'disabled', etc.). The skin templates 27 may be stored as files referenced by the UI control elements 22. Designers may modify or replace a skin templates 27 as desired.

Each UI control's instructions (associated data and function members), which define the control's intrinsic behaviors, may be supplied via script 28 (as opposed to native code in the viewer being the alternative), which is indirectly associated to the UI control element 22 via the initialization function 21. For example, instructions (or scripts) 28 may include toggling a button's state between 'up' and 'down' when clicked upon with the mouse, or translating (panning) the contents of the document within the display window when the thumb of a scrollbar is dragged with the mouse.

The initialization function 21 associates each UI control element 22 to the script that creates that particular element—both the visual front-end (as SVG elements in a document object model (DOM)) and the back-end (an object in memory, with data and function members). The initialization function 21 contains instructions for traversing each node in the DOM and for searching and calling functions associated with UI control elements 22 having names following the predetermined naming convention.

An alternative UI control system comprises the initialization file 21, the skin templates 27 and the scripts 28 of the UI control system 20. The collection of UI control elements are provided independently from the alternative DOM manipulation system as markup syntax for a designer (or developer, or any user) to use when modifying an XML file, such as an SVG document.

The following is an example of a UI control element 22:

```

5      <dsvg:button
        id="myRectangularRadioButton"
        xlink:href="dsvg/skinButton_Default.svg#skinButton"
        label="Click me!"
        width="50"
        height="40"
        x="10"
        y="20"
10     toggle="true"
        group="myRadioButtons"
        checked="true"
        labelX="0"
        labelY="-3"
15     />

```

The 'button' element defines a UI control that can be clicked to trigger an action. The 'button' element may be a push button, a checkbox (i.e., a sticky/toggle button) or a radiobutton, depending on the attributes of the element.

20 The 'button' element contains the name 23:

dsvg:button

which follows the naming convention of adding "dsvg:" as a prefix (associated to a particular namespace earlier in the file) to the element name, so that the viewer's XML parser will allow these non-standard elements in the DOM. As will be described below, 25 this naming convention will also assist the system 20 to search for UI control elements 22, i.e., all UI control elements 22 having this prefix in their name 23. The 'button' element also contains common attributes 24 (or standard attributes) such as 'id', 'label', 'x', 'y', 'width', 'height, etc. The 'button' element also contains the skin template reference attribute 'xlink:href' which is also common to UI control element 22. The attributes 30 common to all UI control elements 22 will be described below.

The 'toggle' attribute specifies whether the button is a toggle/sticky button or not, i.e. whether the button toggles between 'up' and 'down' states when clicked on or instead switches to the 'down' state when the mouse button is pressed and then back to the 'up' state when the mouse button is released. The 'group' attribute is the name of the group to which the button belongs. If the 'group' attribute is provided, and toggle="true", then this results in the functionality of a radiobutton, where selecting any of the radiobuttons belonging to that group results in the other radiobuttons in that group becoming unselected. The 'checked' attribute specifies whether the button is down/checked/selected (true) or up/unchecked/unselected (false). The default is set to "false". The 'labelX' attribute specifies the x-coordinate of the label relative to the 'y' attribute. If the 'labelX' attribute is not provided, the label's position is determined by the skin 27. The 'labelY' attribute specifies the y-coordinate of the bottom edge of the label, relative to the 'x' attribute. If the 'labelY' attribute is not provided, the label's position is determined by the skin 27.

Other example of UI control elements 22 will be described below.

The following is the syntax for the 'button' skin template 27:

```

<?xml version="1.0"?>
<svg>
  <g id="skinButton">
    <g id="label">
      <text x="" y="" style=""> </text>
    </g>
    <g id="up" display="all">
  25    </g>
    <g id="down" display="none">
    </g>
    <g id="hit" display="none">
    </g>
  30    <g id="hover" display="none">
    </g>
  
```

```

5      <g id="focusUP" display="none">
        </g>
        <g id="focusDown" display="none">
          </g>
          <g id="disabledUp" display="none">
            </g>
            <g id="disabledDown" display="none">
              </g>
            </g>
10    </svg>

```

As can be seen in the skin template shown above, a separate group is defined for each possible state of the UI control, where the specific state is identified via the 'id' attribute. The 'display' attributes are not actually required, as the button's associated script sets them itself, but by making the 'up' group visible and the other invisible, it allows the 'up' state to be easily viewed separately in a viewer, so that the designer can easily get a feel for what the button looks like.

The following is the syntax of an example of a 'button' skin:

```

20    <?xml version="1.0"?>
    <svg>
      <g id="skinButton">
        <g id="label">
          <text x="50" y="13"
25    style="font-family:'Tahoma';stroke:none;font-size:12;text-align:
          chor:middle"> </text>
          </g>
          <g id="up" display="all">
            <rect x="0" y="0" width="100" height="18"
30    style="fill:rgb(230,230,230);stroke:none"/>
            <polyline points="0,18 0,0 100,0"
            style="stroke-width:1;stroke:white;fill:none"/>
            <polyline points="0,18 100,18 100,0"
            style="stroke-width:1;stroke:black;fill:none"/>

```

```

    </g>
    <g id="down" display="none">
        <rect x="0" y="0" width="100" height="18"
5      style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0"
      style="stroke-width:1;stroke:black;fill:none"/>
        <polyline points="0,18 100,18 100,0"
      style="stroke-width:1;stroke:white;fill:none"/>
    </g>
10    <g id="hover" display="none">
        <rect x="0" y="0" width="100" height="18"
      style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0"
      style="stroke-width:2;stroke:white;fill:none"/>
15    <polyline points="0,18 100,18 100,0"
      style="stroke-width:2;stroke:black;fill:none"/>
        <rect x="1" y="1" width="98" height="16"
      style="fill:none;stroke:#199DBF"/>
    </g>
20    <g id="disabled" style="opacity:0.5"
      display="none">
        <rect x="0" y="0" width="100" height="18"
      style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0"
25    style="stroke-width:1;stroke:white;fill:none"/>
        <polyline points="0,18 100,18 100,0"
      style="stroke-width:1;stroke:black;fill:none"/>
    </g>
  </g>
30 </svg>

```

As can be seen in the sample skin file above, the label uses the Tahoma font, with a size of 12, and its middle appears 50 units to the right and 13 units below the top-left corner of the button, as defined by its 'x' and 'y' attributes. Its states are drawn using light-gray rectangles and black and white polylines, and the 'hover' state (which appears when the

mouse cursor is overtop of the button) includes an extra light-gray rectangle with a blue outline/stroke.

Other examples of UI control elements 22 will be described below.

The following is an example of SVG markup used in association with an  
 5 embodiment of the UI control system 20.

```

<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
"http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/svg-20000303-st
10 ylable.dtd">
<svg xmlns:dsvg="http://corel.org/dsvg" onload="dsvgInit(evt)"
width="256" height="256">

  <script type="text/ecmascript" xlink:href="dsvg/dSVG.js"/>
  15 <script type="text/ecmascript" xlink:href="dsvg/BaseUI.js"/>
    <script type="text/ecmascript" xlink:href="dsvg/button.js"/>
    <script type="text/ecmascript"
xlink:href="dsvg/constraint.js"/>

  20 <!-- Create a grey background. -->
    <rect x="0" y="0" width="512" height="512"
style="fill:rgb(230,230,230)"/>

    <!-- Create various types of buttons, with no associated
25 behaviours. -->
    <dsvg:button id="myPushButton" label="Click me!" x="50.5"
y="10.5" xlink:href="dsvg/skinButton_Windows.svg#skinButton"/>
    <dsvg:button id="myCheckbox" label="Check me!" x="50.5"
y="50.5" toggle="true"
30 xlink:href="dsvg/skinCheckbox_Default.svg#skinCheckbox"/>
    <dsvg:button id="myRadioButton1" selected="true" label="Check
me!" x="50.5" y="100.5" toggle="true" group="myGroup"
checked="true"
xlink:href="dsvg/skinRadioButton_Default.svg#skinRadioButton"/>

```

```

    <dsvg:button id="myRadioButton2" label="No, me!" x="50.5"
y="125.5" toggle="true" group="myGroup"
xlink:href="dsvg/skinRadioButton_Default.svg#skinRadioButton"/>
    <dsvg:button id="myBigPushButton" label="Look how big I am!"
5 x="50.5" y="150.5" width="150" height="100"
xlink:href="dsvg/skinButton_Windows.svg#skinButton"/>
</svg>

```

Figure 3 is a node representation of the DOM of the above button-01.svg 30. The  
 10 DOM contains a node for the root 'svg' element 31. The DOM representation also  
 contains script nodes 32 and a rect (rectangle) node 33 and UI control element nodes 34  
 for buttons (button). The <dsvg:button> UI control element 34 contains the prefix  
 "dsvg:" in the name 23.

Figures 4A to 4D are screen shots of the file button-01.svg. Figure 4A is a screen  
 15 shot of the file button-01.svg in its initial state, with the mouse cursor not over any  
 buttons. Figure 4B is a screen shot of the display after the user has clicked on the  
 checkbox, clicked on the bottom radiobutton, and is now hovering over the top push-  
 button. Figure 4C is a screen shot of the display when the user is pressing the mouse  
 button down on the top button. Figure 4D is a screen shot of the display when the user  
 20 has let go of the mouse button and moved it away from the UI controls, leaving the top  
 button in its 'focus' state (since it is the last-used UI control). Since the skin template  
 used by the top button does not have an appearance defined for the 'focus' state, the  
 'hover' state's appearance is used instead.

Figure 5 shows an example of a method of creating UI controls of a web  
 25 application at load time (50) in accordance with the UI control system 20. The  
 initialization function 21 is executed/run by the viewer's script interpreter, in response to  
 an "onload" event that the viewer generates when it loads an initial file. This can be  
 achieved by adding an onload="dsvgInit()" attribute to the root 'svg' element. The  
 method (50) begins with the initialization function 21 searching the DOM (beneath the  
 30 root 'svg' element) for a designated UI control element (51). A designated UI control  
 element is a UI control element with a predetermined prefix (which is associated to a  
 particular namespace using the 'xmlns' attribute of the root 'svg' element) to the name of

the name of the UI control element in the DOM. Whether or not the first DOM element is a designated UI control element is determined (51). If the first element is a designated UI control element (52) (e.g., `dsvg:button`), then a function name associated with the designated UI control element is generated (53) and the function is called (54). If a designated UI control element is not found (52), or after a generated function is called (54), the method determines if there are more elements in the DOM to search (55). If there are more elements in the DOM (55), the initialization function determines if the next sibling element is a designated element (56). Steps (52) to (56) are repeated until all elements in the DOM are searched. Once there are no more elements in the DOM to search (55), then the method is done (57).

The method described in Figure 5 will be described using the examples of Figures 2, 3, and 4A to 4D. The viewer's script interpreter will execute the `dsvgInit()` function at load time (i.e., after the DOM has been built and the scripts retrieved and loaded into memory), which will traverse each node in the DOM (56), searching for elements whose names begin with the "dsvg:" prefix (i.e., searching for designated UI control elements 22). The `<dsvg:button>` node is found. Using *a priori* knowledge of the naming conventions for elements and functions, the string "`dsvgButton(element, evt)`" is created, turned into a function and called. This function finds all of the attributes of the 'dsvg:button' element, whose corresponding DOM object was passed in as the parameter 'element', and retrieves their values using the `getAttribute()` DOM API. It then creates a new instance of the button class—creating an object in memory, which contains data and function methods that provide the instructions for how to build its visual front-end and how to behave in response to events generated by the user. Using the value of the 'xlink:href' attribute 26, it retrieves the skin template 27 (which may be in a separate file) and uses it to create the visual front-end of the button by creating the SVG elements in the DOM, in accordance with the skin template 27, and modifying them using some of the other attributes 24, such as 'x', 'y', 'width', 'height', 'label', etc. Other attributes, such as 'toggle' are used to set up how the button will behave and react to user input, which requires that appropriate event listeners be placed (using the `addEventListener()` DOM API) on each group containing the visual appearance of each state, so that the user can interact with the button. Once the `dsvgButton()` function has completed its instructions,



the initialization function 21 then looks for more designated UI control elements 22. It finds four more designated UI control elements 22 and handles them in the same manner as it handled the first. After that, the initialization function 21 does not find any more designated UI control elements 22, and so its job is finished.

5       When the user positions the mouse pointer over any of the buttons (Figures 4A to 4B), the viewer creates an “onmouseover” event, which the event listener hears and dispatches to the button class’s appropriate handler function. If the button was in an ‘up’ state, this function sets the button’s state to be the ‘hover’ state, causing the visual appearance to change to the appearance associated with the ‘hover’ state. Since only the  
10   ‘g’ element containing the associated appearance for the ‘up’ state has an ‘onmouseover’ event listener, only the ‘up’ state will respond to that event by putting the button in the ‘hover’ state. If the button is in the ‘hover’ state and the user moves the mouse pointer away from the button, the “onmouseout” event is generated and passed to its handler function, which changes the button’s state back to whatever it was before it was changed  
15   to the ‘hover’ state.

When the user clicks the mouse pointer down over any of the buttons (Figures 4A to 4B), the viewer creates an “onmousedown” event, which the event listener hears (if the button is not in a ‘disabledUp’ or ‘disabledDown’ state, as the appearances associated with these states have no “onmousedown” event listeners) and dispatches to the button  
20   class’s appropriate handler function. If the button is a toggle-button (i.e., if its ‘toggle’ attribute equals ‘true’), then the “onmousedown” handler function sets the button’s state to be ‘down’ if it was in the ‘up’ state (previous to the ‘hover’ state) or sets it to be ‘up’ if it was in the ‘down’ state (previous to the ‘hover’ state), which causes the visual appearance to change appropriately. If the button is a radio-button (it’s a toggle-button  
25   but also has a ‘group’ attribute defined), then the “onmousedown” handler function will be called, if the button was in the ‘up’ state, causing the button to switch to the ‘down’ state, causing its visual appearance to change appropriately and also causing whichever button belonging to the same radiogroup that was previously in the ‘down’ state to switch to the ‘up’ state. If the button is a push-button (i.e., its ‘toggle’ attribute equals ‘false’),  
30   then the “onmousedown” handler function sets the button’s state to be ‘down’ if it was in the ‘up’ state (previous to the ‘hover’ state). When the user lets go of the mouse button,

the viewer generates the "onmouseup" event, whose handler function sets the button back to the 'up' state.

If a toggle-button's state changes from 'up' to 'down' or from 'down' to 'up', or a push-button's state changes from 'up' to 'down' and then 'up' again, then the  
 5 processActions() handler function is called, which searches all the children of the 'button' element, looking for any elements that begin with the "dsvg:" prefix. It does not find any, and so its job is done.

In the example described above, the function was dynamically generated, i.e., a string was created, having the same prefix as the designated element (without the colon)  
 10 and the same name as the designated element (except with the first letter capitalized) and with the designated element's object and the trigger event object passed in as two parameters. The skin template 27 and the script 28 or set of instructions for the operations of the generated function is stored in a predetermined format either in the document text file or in a separate text file on a file system or webserver, and is loaded  
 15 into memory by the viewer at load time. Alternatively, the initialization function may search for elements that begin with the "dsvg:" prefix and, using an 'if' or 'switch' statement, determine the appropriate predetermined function to call, which again are expected to have been already loaded in memory by the viewer.

It is advantageous, though, for the function names to be generated dynamically, so  
 20 that the main script file containing the initialization function 21 does not need to be updated whenever a new type of UI control element 22 has been created and is available for use.

As well, while the functions 28 that handle each type of UI control element 22 could be stored all in one file, it is advantageous to store them in separate files and  
 25 reference them in the document only if their corresponding UI control element 22 is being used, so that only the code that is required is actually transmitted.

In order for UI control elements 22 to execute desired actions, behavior elements may be inserted as children of the UI control elements 22 (the observer elements). The behavior element will be executed sequentially for each behavior element whose 'event'  
 30 attribute's value matches the observer element's event (e.g., onmouseover, onclick, etc.). If the 'event' attribute is not provided, 'onclick' will be assumed. In the example below,

clicking on the 'Red' button will colour the circle red, while clicking on the 'Blue' button will colour the circle blue.

```

5      <button id="buttonRed" x="10" y="10" label="Red"
      xlink:href="#skinButton">
          <setAttribute elementID="myCircle" name="fill"
value="red"/>
      </button>

10     <button id="buttonBlue" x="10" y="40" label="Red"
      xlink:href="#skinButton">
          <setAttribute elementID="myCircle" name="fill"
value="red"/>
      </button>

15     <circle id="myCircle" x="100" y="10" r="5"
      fill="green"/>

```

Alternatively, the behavior elements can (optionally) be grouped as children of an

20 <action> element, which can be hooked up to the observer element using a <listener> element. For example:

```

      <button id="buttonRed" x="10" y="10" label="Red"
      xlink:href="#skinButton"/>
25     <button id="buttonBlue" x="10" y="40" label="Red"
      xlink:href="#skinButton"/>

      <circle id="myCircle" x="100" y="10" r="5"
      fill="green"/>

30     <action id="setRed">
          <setAttribute elementID="myCircle" name="fill"
value="red"/>
      </action>

35

```

```

<action id="setRed">
    <setAttribute id="setBlue" elementID="myCircle"
name="fill" value="blue"/>
</action>

<listener event="onclick"
observerElementID="buttonRed" handlerID="setRed"/>
<listener event="onclick"
observerElementID="buttonBlue" handlerID="setBlue"/>

```

5

10

Figure 6 shows an example of a method of creating UI controls for a web application in response to an event (60) in accordance with the UI control system 20. The UI control system 20 is built on top of an event-driven architecture, such as SVG, and XML. Once an event occurs on an SVG element (i.e., the observer element), the method (60) begins with passing the event object to a handler function (61). The handler function determines if the first child element of the SVG element associated with the object is a designated element (62). If a designated element is found (63), then the handler function determines if the event attribute 24 of the designated element is equal to the event that has occurred (64). If the event attribute 24 of the designated element is equal to the event which triggered this method (60), then the name of the function associated with the designated element is automatically generated (65) (in accordance with a predetermined function naming convention) and called (66). Preferably, the predetermined function naming convention is similar to the predetermined element naming convention. If a designated element is not found (63), or if the event attribute 24 of the designated element does not match the trigger event (64), or after a generated function is called (66), the event handler determines if there are more child elements of the observer element to search (67). If there are more child elements of the observer element (67), the event handler determines if the next child is a designated element (68). Steps (63) to (68) are repeated until all child elements of the observer element are searched. Once there are no more child elements to search (67), then the handler function is done (69).

30

The initialization file 21 may also search for attributes in elements that are not UI control elements 22. Scripts 28 may be created and associated with the 'dsvg' attribute in

the same manner as with UI control elements. Script functions 28 for 'dsvg' attributes only operate on the object associated with the existing element to which a 'dsvg' attribute is added. A designer may add the 'dsvg' attribute in an SVG file, or any other XML file to be parsed by the viewer 13.

5           Figure 7 shows another example of an method of controlling UI features of a web application (70), in accordance with the UI control system 20. After a user (or designer) marks up an SVG file using the markup syntax of the UI control system and the SVG file is loaded into a viewer 13, the viewer 13 creates an "onload" event which is received by an <svg> element. The method (70) begins with the initialization function 21. A  
 10   dsvgInit() initialization function 21 is called (71) by the viewer's script interpreter, which traverses the nodes of the DOM of the SVG file. The initialization function 21 determines if the first DOM element is a designated element (72). If a designated element is found (73) and the 'event' attribute of the designated element is set to "onload" (74), then the name of the function associated with the designated element is automatically  
 15   generated (75) (in accordance with a predetermined function naming convention) and called (76). Preferably, the predetermined function naming convention is similar to the predetermined element naming convention. If a designated element is not found (73), the initialization function 21 determines if the regular SVG element contains any designated attributes (77) (which begin with the "dsvg:" prefix). If any designated attributes are  
 20   found (77) (e.g., dsvg:toolTip="#skinTooltip\_traditional"), then the names of the functions associated with the designated attributes are automatically generated (78) (again, in accordance with a predetermined function naming convention) and called (79).

          If a designated attribute is not found (77), then the initialization file 21 determines if the regular SVG element has any child elements (80). If the regular SVG element has a  
 25   child element (80) and the child element is a designated element 22 (81), then the initialization file 21 determines the value of the designated element's 'event' attribute (i.e., the event that will trigger the execution of the designated element's associated function) and adds that event listener to the parent SVG element (82) (via the  
 30   addEventListener() DOM API). If the child element is not a designated element 22 (81), then the initialization file 21 determines if there are any other children of the regular SVG element (83). If there are more children (83), then the initialization file searches the next

child of the regular SVG element (84). Steps (81) to (84) repeat until there are no more children of the regular SVG element.

If there are no more children of the regular SVG element (83), or after a generated function is called (76, 79), or if the event attribute of a designated element is not equal to "onload" (74), or there are no more child elements in a regular SVG element to search (80), the initialization file 21 determines if there are more elements in the DOM to search (85). If there are more elements in the DOM (85), the initialization file determines if the next sibling element is a designated element (86). Steps (83) to (86) are repeated until all elements in the DOM are searched. Once there are no more elements in the DOM to search (85), then the initialization function 21 is done and the viewer 13 waits for an event to occur (87).

Once an event occurs on an SVG element (i.e., the observer element), that event object is passed to any handler function with which it has been associated (88). The handler function changes the UI control's state (89) to whatever it should be, according to the logic inherent in the specific UI control, which causes the visual appearance to change (the 'g' element containing the SVG for the previous state gets its 'display' attribute set to 'none', while the 'g' element containing the SVG for the new state gets its 'display' attribute set to 'all'). If the UI control's logic dictates that this change in state requires that any associated behavior elements be run (90), then the event object is passed to the processActions() handler function (91). The processActions() handler function traverses all children of the UI control element and performs any action required (92). If the UI control's logic does not dictate that the change in state requires any associated behavior elements to be run (90) or after the processActions() handler function has performed the required actions (92), then the event handler function is done and the viewer waits for another event to occur (87).

### Referencing Attributes

To create an application, a designer often desires to reference the current value of another element's attributes. The system 20 allows for the following syntax to perform this reference:

%frameID.objectID.docID.elementID.childElementID@attributeName%

If not specified, the 'frameID', 'objectID' and 'docID' are assumed to be the current frame, object and document. In the following example, checking and unchecking the checkbox will disable and enable the textbox.

```

10  <button id="myCheckBox" x="10" y="10" toggle="true"
    label="Disable slider" xlink:href="#skinCheckbox">
    <setAttribute elementID="myTextBox"
    name="disabled" value="%myCheckBox.checked%"/>
    </button>

```

```

    <textBox id="myTextBox"/>

```

15 The system 20 also allows for parenthesis and mathematical operators within the %% expression. For example, in the markup:

```

20  <dsvg:button id="button" x="50" y="50" label="foo"
    toggle="true" group="pickPanGroup"
    xlink:href="skinButton_Windows.svg#skinButton"
    selected="true" />

    <dsvg:button id="button_2" x="200" y="200" label="foo"
    toggle="true" group="pickPanGroup"
    xlink:href="skinButton_Windows.svg#skinButton"
25  selected="true" />

```

an attribute "foo %button\_(button@x - 48)@x + 14 \* button@y% bar%button@x%" is parsed as follows:

```

30  'foo %button_(button@x - 48)@x + 14 * button@y% bar%button@x/2%'
    -> 'foo %button_(50 - 48)@x + 14 * button@y% bar%button@x/2%'

```

```

-> 'foo %button_(2)@x + 14 * button@y% bar%button@x/2%'
-> 'foo %button_2@x + 14 * button@y% bar%button@x/2%'
-> 'foo %200 + 14 * 50% bar%button@x/2%'
-> 'foo %200 + 700% bar%button@x/2%'
5  -> 'foo 900 bar%button@x/2%'
    -> 'foo 900 bar25'

```

Figure 8 shows an example of a method of creating an element for controlling user interface features of a web application (200) in accordance with the UI control system 20.

10 The method (200) begins with categorising low level UI controls into abstractions or fundamental core UI control groupings (201). Next, common attributes of a core UI control grouping are determined (202). Next, other attributes 25 of the core UI control grouping are determined (203). Next, a skin template for the core UI control grouping is created (204). Next, a core UI control element 22 having the common attributes 24 and 26, other attributes 25 is create along with a skin template 27 of the UI control

15 grouping (205). The method is done (206).

Figure 9 shows an example of a method of creating a plurality of elements for controlling user interface features of a web application (210) in accordance with the UI control system 20. The method (210) begins with organizing low level UI control

20 elements 22 into groupings of similar UI control elements (211). Next, core UI control names are designated to the groupings (212). Next, variations of a grouping are analyzed to determine common attributes of the grouping (213). A list of common attributes used to describe the variations is compiled (214). Next, fundamental states of the core UI control grouping are determined (215). These fundamental states include different

25 appearances that supply the intrinsic behavior of the core UI control (e.g. a UI control in a 'disabled' state does nothing when clicked upon). The fundamental states include "up", "down", "hover", "hit", "focusUP", "focusDown", "disabledUp" and "disabledDown". Once the fundamental states are determined (85), a method of allowing for absolute positioning of UI controls is determined (216). For example, providing x and y co-

30 ordinates. Next, a method of allowing for absolute customization of appearance of the



core UI control is determined (217), so that the appearance of the UI controls is in no way determined by the viewer or by the script that creates the UI controls. This step (217) goes beyond a simple modification of styling properties, which would allow you to specify the UI control's fill-colour or stroke-colour, for instance. Instead, the appearance for each possible state is defined within a skin template, whose location is defined by the 'xlink:href' attribute, in the same manner in which the 'xlink:href' is typically used within regular SVG markup (e.g., xlink:href="dsvg/skinButton\_Default.svg#skinButton" means that the skin template is an element whose 'id' attribute is "skinButton" within the file skinbutton\_Default.svg within the dsvg directory). Within the skin template 27, a 'g' element (group container element) can exist for each possible state, within which any SVG markup is allowed (note that if the UI control system 20 is used within an SVG viewer, the skin template must contain SVG markup; if the UI control system 10 is used within a different viewer, the markup within the skin template must conform to the markup that the viewer expects). Thus the UI control's back-end (the script object, with data and function members) has no pre-conceived notions of what the UI control actually looks like for its various states. This poses problems for resizing the UI controls based on the 'width' and 'height' attributes, however. It is a simple matter to compare the UI control's width and height with the desired width and height, and from that calculate the scaling that is required in the x- and y-directions, and then apply those scale factors to the UI control's visual front-end (a modified version of the skin template) via a transformation (in SVG, using the 'transform' attribute). But that causes the UI control to not only change dimensions, but it also stretches or squishes all the visual elements and changes their absolute positions with respect to one another--the label may be stretched or squished and may be further from or closer to the UI control than originally intended, the stroke-width's may be thicker or thinner, the button in a comboBox, may be stretched, etc. The solution is to give the designer the capability of defining the constraints, directly in the skin template, that will provide the detailed information needed to "intelligently" resize the UI control. For example, one might specify that the label always remain unscaled and that its position always be 3 pixels above the UI control, or that the the stroke-width always be equal to 1, or that the button of a comboBox always has its width automatically adjusted so as to preserve its original aspect ratio.

Even if instead of applying the scale factors to the UI control via a transformation, the scale factors were instead used to directly modify all of the coordinates for all of the various types of elements used for the UI control's appearance (e.g., rect, circle, path, line, polyline, etc.), that could still result in some of the same problems. For instance, the designer might intend for the label's position and/or size to scale or not. Thus the constraints must still be defined within the skin template. Once all of the above determinations have been performed, other attributes that the core UI control element needs are determined (218). A core UI control element 22 is then created. A name is assigned to a core UI control element 22 pursuant to a naming convention (219). Next the common attributes 24, of the UI control grouping are assigned to the core UI control element 22 (220). The other attributes 25 of the UI control grouping are then assigned to the core UI control element 22 (221). Finally, the skin template reference 26 is assigned to the core UI control element 22 (222). The skin template 27 comprises descriptions of the visual appearance of each state for the UI control. The set of instructions (i.e., the script) 28 that creates the UI control's back-end (i.e., the object in memory with data and function members, which determine how to construct the UI control using the skin template 27, what intrinsic behaviors to exhibit and how to interact with the user) are not directly associated to the UI control element 22, but rather are indirectly associated to the UI control element 22 via the initialization function 21 (i.e., the UI control element 22 does not actually reference the script function 28 or the script file 28 that creates and controls the UI control). The core UI control element's 22 set of instructions (script) 28 may be stored in an independent file. Once all core UI control elements 22 have been created (223), the method is done (224). Other steps may be added to the method (210). Default settings may be initiated for the core UI control elements 22, if desired.

There are many advantages to the UI control system 20. The UI control system 20 encompasses general UI controls desired to build web applications. The UI control system 20 allows for absolute positioning, absolute control of the appearance, and may easily be hooked up to actions, defined via markup or script functions, that affect any element in a DOM. The UI control system 20 may be used for forms. However, the control system 20 is not tied to a model specifically intended for forms (i.e., does not

force the author to jump through hoops to create an application which does not use forms).

The UI control system 20 assists web designers with no programming skills to create dynamic, interactive web applications. It also aids experienced programmers to  
5 create dynamic, interactive web applications much more easily and rapidly. Because the UI control system 20 involves an XML markup language (as opposed to just script functions), the attributes and data and even the elements themselves can be made to be data-driven at run-time, using (at design-time) existing or new software that allows one to visually map input XML markup to output XML markup, resulting in an extensible  
10 stylesheet language transformation (XSLT) code (or any other language useful for XML transformations) which will actually modify the UI control elements 20 based on the input XML data/markup.

The UI control system 20 can also be natively-implemented, accessing the exposed DOM API's in the same manner as the script implementation. A native  
15 implementation could be much faster because unlike script, which gets interpreted at run-time, native code (e.g. C++ or C) gets interpreted at compile time and gets optimized by the compiler. The natively-implemented UI control system 20 could also access any unexposed, lower-level object model API's directly, rather than the exposed higher-level DOM API's, which could further improve performance. If natively implemented, the  
20 amount of data needed to be transferred may be greatly reduced, since there is no script that needs to be transmitted, which is especially beneficial for wireless devices with low bandwidth and small memory. Using a markup language for the UI control elements 22 is also beneficial because it allows for the possibility of further reducing the file size by creating a binary version of the markup language that employs opcodes—predetermined  
25 arrangements of bits (1's and 0's) that correspond to particular element names and attributes. Unlike textual markup, which must be parsed (compared to predetermine strings/text to establish the meaning of the text) in order to create the DOM, binary opcodes can be compared to identical binary opcodes, which is much faster than string comparisons, in order to build the DOM much faster.

The UI control system 20 according to the present invention may be implemented by any hardware, software or a combination of hardware and software having the above described functions. The software code, either in its entirety or a part thereof, may be stored in a computer readable memory. Further, a computer data signal representing the software code which may be embedded in a carrier wave may be transmitted via a communication network. Such a computer readable memory and a computer data signal are also within the scope of the present invention, as well as the hardware, software and the combination thereof.

While particular embodiments of the present invention have been shown and described, changes and modifications may be made to such embodiments without departing from the true scope of the invention.

#### List of UI Control Elements, Attributes, and Skin Templates

Some examples of UI control elements 22, common attributes 24 and 26, skin templates 27 and examples of skins, in accordance with the UI control system 20, are provided below. Other UI control elements, attributes and skins may be created. The provided UI control elements, attributes and skins are examples of one implementation of a DOM manipulation markup language.

#### Attributes Common to All UI Controls (24 and 26)

**xlink:href="<uri>"**

A reference to the skin's parent element, stored either internally in the <defs> block, or in an external file. (The skin template reference attribute 26).

**id="name"**

Standard XML attribute for assigning a unique *name* to an element.

**label="string"**

The text associated with the UI control, the position of which is determined by the skin, unless overridden by the 'labelX' and 'labelY' attributes.

**x="<coordinate>"**

Specifies the x-coordinate of the left edge of the UI control. If not provided, the default is zero.

**y="<coordinate>"**

5 Specifies the y-coordinate of the top edge of the UI control. If not provided, the default is zero.

**width="<length>"**

Specifies the width of the UI control. If not provided, the width is determined by the skin.

10 **height="<length>"**

Specifies the height of the UI control. If not provided, the height is determined by the skin.

**preserveAspectRatio="(true | false)"**

15 If 'width' or 'height' is given, the other dimension is automatically calculated so as to preserve the UI control's aspect ratio. If both 'width' and 'height' are given, this attribute is ignored.

**labelX="<coordinate>"**

20 Specifies the x-coordinate of the left edge of the label, relative to the 'y' attribute. If not provided, the label's position is determined by the skin. Note: It is currently relative to wherever the skin specifies the label to be, but this will be changed.

**labelY="<coordinate>"**

25 Specifies the y-coordinate of the bottom edge of the label, relative to the 'x' attribute. If not provided, the label's position is determined by the skin. Note: It is currently relative to wherever the skin specifies the label to be, but this will be changed.

**disabled=(true | false)**

Specifies whether the UI control is disabled (true) and cannot be used, or not (false).

**state\_hover="name"**

5 Specifies the ID of the UI control's <state> child element, to override the appearance of the "hover" state, as defined in the skin.

**state\_focus\_up="name"**

Specifies the ID of the UI control's <state> child element, to override the appearance of the "focus\_up" state, as defined in the skin.

**state\_focus\_down="name"**

10 Specifies the ID of the UI control's <state> child element, to override the appearance of the "focus\_down" state, as defined in the skin.

**state\_up="name"**

Specifies the ID of the UI control's <state> child element, to override the appearance of the "up" state, as defined in the skin.

15 **state\_down="name"**

Specifies the ID of the UI control's <state> child element, to override the appearance of the "down" state, as defined in the skin.

**state\_hit="name"**

20 Specifies the ID of the UI control's <state> child element, to override the appearance of the "hit" state, as defined in the skin.

**state\_disabled\_up="name"**

Specifies the ID of the UI control's <state> child element, to override the appearance of the "disabled\_up" state, as defined in the skin.

**state\_disabled\_down="name"**

25 Specifies the ID of the UI control's <state> child element, to override the appearance of the "disabled\_down" state, as defined in the skin.

UI Control Elements 22**The 'button' element**

The 'button' element defines a control that can be clicked to trigger an action. It can be a push button, a checkbox (same as a sticky/toggle button) or a radiobutton, depending on its attributes.

*Attribute definitions:*

**%standard attributes%**

The standard attributes, such as id, label, etc.

**toggle="(true | false)"**

Specifies whether the button is a "toggle" or "sticky" button, meaning that it toggles between "up" and "down" states when clicked on. A checkbox has toggle="true".

**group="string"**

The name of the group that the button belongs too. If provided, and toggle="true", this results in the functionality of a radio button.

**checked="(true | false)"**

Specifies whether the button is down/checked (true) or up/unchecked (false). The default is 'false'.

**labelX="<coordinate>"**

Specifies the x-coordinate of the left edge of the label, relative to the 'y' attribute. If not provided, the label's position is determined by the skin.

**labelY="<coordinate>"**

Specifies the y-coordinate of the bottom edge of the label, relative to the 'x' attribute. If not provided, the label's position is determined by the skin

**The 'calendar' element**

The 'calendar' element defines a calendar control, displaying the specified month of the specified year.

*Attribute definitions:*

**year="integer"**

5                    Specifies the year for the calendar to display.

**month="integer"**

                    Specifies the month for the calendar to display.

**selectable="(all | weekdays | weekends)"**

                    Specifies what days can be selected by the user.

10                   **multiSelect="(true | false)"**

                    Specifies whether the user can select multiple days (true) or not (false).

### The 'comboBox' element

15                   The 'comboBox' element defines a comboBox control, used to display a list of items, from which only one can be selected. It is a combination of a textbox and a listbox. There are three types:

                    An editable textbox with a drop-down listbox.

                    An editable textbox with a fixed-size listbox.

                    A non-editable drop-down listbox.

20                   *Attribute definitions:*

**%standard attributes%**

                    The standard attributes, such as id, label, etc.

**dropdown="(true | false)"**

                    Specifies whether the comboBox has a dropdown list (true).

25                   **editable="(true | false)"**



Specifies whether the comboBox is editable (true) or not (false). If editable, typing text and pressing Enter results in a new <item> element being created as a child of the <comboBox>, with its 'value' attribute set to the text just entered.

5      **value="string"**

The value of the currently selected item's 'label' attribute.

**name="string"**

The value of the currently selected item's 'name' attribute, which is not displayed.

10      **selectedID="integer"**

The value of the currently selected item's 'id' attribute.

#### The 'contextMenu' element

15      The 'contextMenu' element defines a menu that is associated with a particular element or group of elements, which appears at the position of the mouse pointer when right-clicking on an element.

##### *Attribute definitions:*

##### **%standard attributes%**

The standard attributes, such as id, label, etc.

20      **eventSource="name"**

Specifies the ID of the element that triggered the contextMenu to appear. This attribute gets automatically populated for reference purposes. It should never be set via markup.

**value="string"**

25      The value of the currently selected item's 'label' attribute. This represents the initial item selected. When a new item is selected, the 'value' attribute is automatically updated to reflect the 'label' attribute of the selected item.

**name="string"**

The value of the currently selected item's 'name' attribute, which is not displayed.

**selectedID="integer"**

5           The value of the currently selected item's 'id' attribute.

### The 'date' element

The 'date' element defines properties for a particular date for its parent calendar control. It must be a child of a 'calendar' element.

10       *Attribute definitions:*

**id="name"**

Standard XML attribute for assigning a unique *name* to an element.

**year="integer"**

Specifies the year for this particular date.

15       **month="integer"**

Specifies the month for this particular date.

**day="integer"**

Specifies the day for this particular date.

**name="string"**

20           Specifies a hidden string to associate with this particular date (which, unlike 'id', does not need to be unique). e.g. "HOLIDAY".

**selectable="(true | false)"**

Specifies whether this particular date is selectable (true) or not (false).

**label="string"**

25           Specifies the text label to be displayed for this particular date.

## The 'dialog' element

The 'dialog' element defines a window container, specifically for the purpose of collecting data which can be accepted or cancelled. It has inherent Accept/OK, Cancel, Help and Info buttons.

### 5 *Attribute definitions:*

**showTitle="(true | false)"**

Specifies whether to display a titlebar (true) or not (false). If true, the title is obtained from the 'label' attribute.

**showMinimizeButton="(true | false)"**

10 Specifies whether to display the minimize button or not.

**showMaximizeButton="(true | false)"**

Specifies whether to display the maximize button or not.

**showCloseButton="(true | false)"**

Specifies whether to display the close button or not.

15 **showAcceptButton="(true | false)"**

Specifies whether to display the Accept/OK button or not.

**showCancelButton="(true | false)"**

Specifies whether to display the Cancel button or not.

**showHelpButton="(true | false)"**

20 Specifies whether to display the Help button or not.

**showInfoButton="(true | false)"**

Specifies whether to display the Information button or not.

**modal="(true | false)"**

Specifies whether the window is modal (true) or modeless (false).

25 **movable="(true | false)"**

Specifies whether the window is movable (true) by clicking on the titlebar and dragging, or not (false).

**state="(minimized | maximized | normal)"**

5 Specifies the initial state of the window. If the Minimize or Maximize buttons are pressed, this attribute is automatically updated.

### The 'frame' element

The 'frame' element defines a container element.

10 *Attribute definitions:*

**xlink:href="<uri>"**

Specifies the document to display within the 'frame' container.

### The 'item' element

15 The 'item' element defines a particular item of a comboBox, listBox, listView, contextMenu or menuBar. It can only exist as a child of these elements.

*Attribute definitions:*

**id="name"**

Standard XML attribute for assigning a unique *name* to an element.

20 **accessKey="string"**

The shortcut key which, when pressed, selects this item. Usually the first letter of the displayed string. If not unique among the 'item' elements, repeated striking of this key will select the next item with that 'accessKey'.

**label="string"**

25 The displayed text of this item.

**name="string"**

The hidden text of this item. Optional.

**xlink:href="<uri>"**

A reference to the parent element of a graphic, stored either internally in the <defs> block or in an external file, to be displayed for that item.

5      **anything="string"**

Any attribute name at all, in which to store data. e.g. 'time', 'date', 'price', etc. Used only for the 'listView' element.

### The 'listBox' element

10      The 'listBox' element defines a listBox control, used to display a list of items, from which one or many can be selected. Unlike the 'comboBox' element, it cannot be a drop-down list but is instead of a specified size. If all its items cannot be displayed within that size, scrollbars automatically appear.

*Attribute definitions:*

15      **%standard attributes%**

The standard attributes, such as id, label, etc.

**multiSelect="(true | false)"**

Specifies whether more than one item can be selected (true) or not (false).

**editable="(true | false)"**

20      Specifies whether the listBox is editable (true) or not (false). If editable, typing text and pressing Enter results in a new <item> element being created as a child of the <listbox>, with its 'value' attribute set to the text just entered.

**value="string"**

25      The value of the currently selected item's 'label' attribute. This represents the initial item selected. When a new item is selected, the 'value' attribute is automatically updated to reflect the 'label' attribute of the selected item.

**name="string"**

The value of the currently selected item's 'name' attribute, which is not displayed.

**selectedID="integer"**

5           The value of the currently selected item's 'id' attribute.

### The 'listView' element

10           The 'listView' element defines a listView control, used to display a columnar list of items, from which one or many can be selected. Unlike the 'combobox' element, it cannot be a drop-down list but is instead of a specified size. If all its items cannot be displayed within that size, scrollbars automatically appear. The listView control allows for any number of columns—each of which can be visible or not, editable or not, and can have its own formatting options, which include all of the formatting options available to the 'textBox' element. It also allows for  
15           column headers, column resizing (by dragging the column separator borders) and list sorting (by clicking on the header buttons).

#### *Attribute definitions:*

##### **%standard attributes%**

The standard attributes, such as id, label, etc.

20           **multiSelect="(true | false)"**

Specifies whether more than one item can be selected (true) or not (false).

**editable="(true | false)"**

25           Specifies whether the listbox is editable (true) or not (false). If editable, typing text and pressing Enter results in a new <item> element being created as a child of the <listbox>, with its 'value' attribute set to the text just entered.

**display="string"**

A semicolon-delimited list of the names of the attributes in all of the listBox's <item> children that is to be actually displayed, in that order.

***anything="string"***

5 Any attribute name at all, in which to store data. e.g. 'time', 'date', 'price', etc. These attributes must also appear in the listView's child 'item' elements. They represent the initial item selected. When a new item is selected, these attributes are automatically updated to reflect the attributes of the selected item.

***selectedID="integer"***

10 The value of the currently selected item's 'id' attribute, if it has one.

### The 'menuBar' element

The 'menuBar' element defines a container for menu items. The expected children are 'item' elements.

15 *Attribute definitions:*

***%standard attributes%***

The standard attributes, such as id, label, etc.

***value="string"***

20 The value of the currently selected child 'item' element's 'label' attribute. This represents the initial item selected. When a new item is selected, the 'value' attribute is automatically updated to reflect the 'label' attribute of the selected item.

***name="string"***

25 The value of the currently selected child 'item' element's 'name' attribute, which is not displayed.

***hideGrippy="(true | false)"***

Specifies whether to hide the grippy (true) or not (false).

### The 'menuSeparator' element

The 'menuSeparator' element defines a separator to be displayed between 'item' elements. It can only exist as a child of comboBox, listBox, listView, contextMenu or menuBar.

5 *Attribute definitions:*

**id="name"**

Standard XML attribute for assigning a unique *name* to an element.

**xlink:href="<uri>"**

10 A reference to the parent element of a graphic, stored either interally in the <defs> block or in an external file, to be displayed for that item. If not provided (which is normally the case), its skin is obtained from the skin of the parent UI control.

### The 'messageBox' element

15 The 'messageBox' element defines a simple dialog with a message and a boolean response buttons. It can have action elements as children, but any graphical child elements will be ignored.

*Attribute definitions:*

**showTitle="(true | false)"**

20 Specifies whether to display a titlebar (true) or not (false). If true, the title is obtained from the 'label' attribute.

**showYesButton="(true | false)"**

Specifies whether to display the Yes button or not.

**showNoButton="(true | false)"**

25 Specifies whether to display the No button or not.

**m vable="(true | false)"**



Specifies whether the window is movable (true) by clicking on the titlebar and dragging, or not (false).

**label="string"**

Specifies the message to display within the message box.

5 **labelYes="string"**

Specifies the label of the Yes button. The default is "Yes".

**labelNo="string"**

Specifies the label of the No button. The default is "No".

## 10 The 'scrollbar' element

The 'scrollbar' element defines the vertical and horizontal scrollbars to be associated with its parent, usually a document or other UI container element, such as a <window> or <frame>.

*Attribute definitions:*

15 **bars="(both | horizontal | vertical | none)"**

Specifies whether a horizontal scrollbar (at the bottom), a vertical scrollbar (on the right) or both should appear when required.

## The 'slider' element

20 The 'slider' element defines a slider control, used to select a value from a range of allowed values by dragging its 'thumb' control.

*Attribute definitions:*

**%standard attributes%**

The standard attributes, such as id, label, etc.

25 **min="integer"**

Specifies the minimum value of the slider.

**max="integer"**

Specifies the maximum value of the slider.

**minPosition="integer"**

Specifies the minimum allowed value of the thumb.

5 **maxPosition="integer"**

Specifies the maximum allowed value of the thumb.

**value="integer"**

Specifies the initial value of the slider. As the slider's thumb is dragged, this attribute is automatically updated.

10 **increment="integer"**

Specifies the allowed values that the slider can create, i.e. only 'min' + multiples of 'increment' are allowed.

**pageIncrement="integer"**

15 Specifies the amount that the thumb should move, in whatever units the slider represents (i.e. not in pixels), when clicking to the left or right of it. If defined as zero, or not defined at all, the thumb will snap to wherever the user clicks.

**orientation="integer"**

20 Specifies the rotation angle of the slider, in degrees clockwise. The default is 0 degrees. Note: this will likely be renamed to be 'rotation'.

**ticksMajor="integer"**

Specifies the interval at which major tick marks (whose appearance is defined in the skin) are displayed, in whatever units the slider represents.

**ticksMinor="integer"**

25 Specifies the interval at which minor tick marks (whose appearance is defined in the skin) are displayed, in whatever units the slider represents.

### The 'spin' element

The 'spin' element defines a spin control, used to select a value from a range of allowed values by pressing the up- or down-buttons, or by entering numbers in its textBox component.

5 *Attribute definitions:*

*%standard attributes%*

The standard attributes, such as id, label, etc.

**min="integer"**

Specifies the minimum value of the spin control.

10 **max="integer"**

Specifies the maximum value of the spin control.

**value="integer"**

Specifies the initial value of the spin control.

**increment="integer"**

15 Specifies the allowed values that the spin control can create, i.e. only 'min' + multiples of 'increment' are allowed.

### The 'state' element

20 The 'state' element defines an alternate skin to be available for the parent UI control to use to override the appearance of any of its states. It must be a child of a UI control element.

*Attribute definitions:*

**xlink:href="<uri>"**

25 A reference to the parent element of a graphic, stored either interally in the <defs> block or in an external file, to be available to be used to override the appearance of a particular state for the parent UI control.

## The 'textBox' element

The 'textBox' element defines a text input field, with any number of rows, in which the user can enter text.

### *Attribute definitions:*

5     **%standard attributes%**

The standard attributes, such as id, label, etc.

**value="*string*"**

Specifies the default text within the textbox. Whenever the textbox loses focus, this attribute is updated to reflect the current contents of the textbox.

10     Modifying this attribute externally, via the <setAttribute> element, should result in the contents being updated appropriately.

**numLines="(integer | auto)"**

Specifies the number of lines allowed in the textbox. If this number of lines cannot be displayed within the textbox, a vertical scrollbar will automatically appear when needed. If numlines="auto", the textbox will display as many lines as can fit within it. The default is "auto".

15

**maxLength="(integer | auto)"**

Specifies the maximum number of characters allowed in the textbox. If that many characters cannot fit within the textbox, the text will pan horizontally so as to allow text entry. If maxLength="auto", the textbox will allow as many characters as can fit within it. The default is "auto".

20

**wrap="(true | false)"**

Specifies whether to auto-wrap text (true) or not (false). If true, text will never scroll horizontally but will instead jump to the beginning of the line below.

25

**readOnly="(true | false)"**

Specifies if the textbox is non-editable (true) or editable (false).

**secret="(true | false)"**

Specifies whether the text is secret (true) or not (false). If secret, such as a password, only asterisks (\*) will be displayed, although the real characters still exist and get copied to the 'value' attribute when the textbox loses focus.

**dataType="(any | alphabetic | alphanumeric | numeric | symbolic | date)"**

Specifies the type of data that is allowed to be entered.

**mask="string"**

Specifies a pattern that allows extra characters to be inserted into the data as it is entered and/or only allows specific characters in specific locations.

**case="(mixed | upper | upperFirstChar | lower)"**

Specifies whether the data should keep its case as entered (mixed), should be set to all uppercase (upper), should have its first character set to uppercase (upperFirstChar), or should be set to lowercase (lower).

## **The 'toolBar' element**

The 'toolBar' element defines a container for buttons.

*Attribute definitions:*

**%standard attributes%**

The standard attributes, such as id, label, etc.

**selectedID="name"**

The value of the currently selected child 'button' element's 'id' attribute.

**hideGrippy="(true | false)"**

Specifies whether to hide the grippy (true) or not (false).

## **The 'window' element**

The 'window' element defines a top-level container element that may be either modal or modeless, and may be moveable or not.

*Attribute definitions:*

*%standard attributes%*

5           The standard attributes, such as id, label, etc.

**showTitle="(true | false)"**

Specifies whether to display a titlebar (true) or not (false). If true, the title is obtained from the 'label' attribute.

**showMinimizeButton="(true | false)"**

10           Specifies whether to display the minimize button or not.

**showMaximizeButton="(true | false)"**

Specifies whether to display the maximize button or not.

**showCloseButton="(true | false)"**

Specifies whether to display the close button or not.

15           **modal="(true | false)"**

Specifies whether the window is modal (true) or modeless (false).

**movable="(true | false)"**

Specifies whether the window can be moved (true) by clicking on the titlebar and dragging, or not (false).

20           **state="(minimized | maximized | normal)"**

Specifies the initial state of the window. If the Minimize or Maximize buttons are pressed, this attribute is automatically updated.

UI Control Attributes

25           The following attributes can be applied to any element, to create a context-sensitive UI control, which may be event-triggered.

**toolTip="`<uri>`"**

- A reference to the parent element of the tooltip's skin, stored either interally in the `<defs>` block, or in an external file. If provided, this element and all child elements containing a 'title' attribute will have a tooltip display when the mouse cursor is over the element, using the text supplied in the 'title' attribute.

**tipDelay="integer"**

Specifies the delay, in seconds, between when the mouse first hovers over an element and when its tooltip appears.

**tipTrack="(true | false)"**

- Specifies whether the tooltip should follow the mouse (true) or stay in one position until the mouse cursor leaves the element (false).

**contextMenu="string"**

Specifies the 'id' of the `<contextMenu>` element that should appear when the user right-clicks on the element containing this attribute.

- nativeContextMenu="string"**

Specifies the 'id' of the `<menu>` element that should appear when the user right-clicks on the element containing this attribute. `<menu>` is an Adobe-specific markup for the Adobe SVG Viewer. Thus this attribute is only for use with the Adobe SVGViewer.

20

#### Skin Templates 27 for UI Controls

#### The 'button' Skin Template

`<?xml version="1.0"?>`

`<svg>`

- 25 `<g id="skinButton">`

`<g id="label">`

`<text x="" y="" style=""> </text>`

```

    </g>
    <g id="up" display="all">
    </g>
    <g id="down" display="none">
5    </g>
    <g id="hit" display="none">
    </g>
    <g id="hover" display="none">
    </g>
10    <g id="focusUp" display="none">
    </g>
    <g id="focusDown" display="none">
    </g>
    <g id="disabled_up" display="none">
15    </g>
    <g id="disabled_down" display="none">
    </g>
    </g>
  </svg>

```

20

### Example 'button' Skin

```

<?xml version="1.0"?>
<svg>
  <g id="skinButton">
25    <g id="label">

```



```

    <text x ="50" y="13"
style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

</g>

<g id="up" display="all">
5    <rect x="0" y="0" width="100" height="18"
style="fill:rgb(230,230,230);stroke:none"/>

    <polyline points="0,18 0,0 100,0" style="stroke-width:1;stroke:white;fill:none"/>

    <polyline points="0,18 100,18 100,0"
style="stroke-width:1;stroke:black;fill:none"/>
10 </g>

    <g id="down" display="none">

    <rect x="0" y="0" width="100" height="18"
style="fill:rgb(230,230,230);stroke:none"/>

    <polyline points="0,18 0,0 100,0" style="stroke-width:1;stroke:black;fill:none"/>
15 <polyline points="0,18 100,18 100,0"
style="stroke-width:1;stroke:white;fill:none"/>

    </g>

    <g id="hover" display="none">

    <rect x="0" y="0" width="100" height="18"
20 style="fill:rgb(230,230,230);stroke:none"/>

    <polyline points="0,18 0,0 100,0" style="stroke-width:2;stroke:white;fill:none"/>

    <polyline points="0,18 100,18 100,0"
style="stroke-width:2;stroke:black;fill:none"/>

    <rect x="1" y="1" width="98" height="16" style="fill:none;stroke:#199DBF"/>
25 </g>

    <g id="disabled" style="opacity:0.5" display="none">

```

```

    <rect x="0" y="0" width="100" height="18"
style="fill:rgb(230,230,230);stroke:none"/>

    <polyline points="0,18 0,0 100,0" style="stroke-width:1;stroke:white;fill:none"/>

    <polyline points="0,18 100,18 100,0"
5 style="stroke-width:1;stroke:black;fill:none"/>

    </g>

    </g>

</svg>

```

## 10 The 'contextMenu' Skin Template

```

<?xml version="1.0"?>

<svg>

    <g id="skinContextMenu">

        <g id="skinContextMenu_top" display="all">

15    </g>

        <g id="skinContextMenu_middle" display="all">

            <text x="" y="" id="label"> </text>

            </g>

            <g id="skinContextMenu_middle_hover" display="none">

20    <text x="" y="" id="label"> </text>

            </g>

            <g id="skinContextMenu_middle_selected" display="none">

                <text x="" y="" id="label"> </text>

            </g>

25    <g id="skinContextMenu_bottom" display="all">

```

```

    </g>
  </g>
</svg>

```

## 5 Example 'contextMenu' Skin

```

<?xml version="1.0"?>

<svg>
  <g id="skinContextMenu">
    <g id="skinContextMenu_top" display="all">
10    <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-width:1" />
        <line x1="150" y1="17" x2="150" y2="20"
style="stroke:#000000;stroke-width:1" />
        <line x1="0" y1="17" x2="150" y2="17" style="stroke:#000000;stroke-width:1"
/>
15    <rect x="1" y="18" width="148" height="2" style="fill:#FFFFFF"/>
    </g>
    <g id="skinContextMenu_middle" display="all">
        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-width:1" />
        <line x1="150" y1="20" x2="150" y2="35"
20 style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="20" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>
        <text x="3" y="32" id="label"> </text>
    </g>
25 <g id="skinContextMenu_middle_hover" display="none">
        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-width:1" />

```

```

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>
5      <text x="3" y="32" id="label"> </text>

      </g>

      <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
10    style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="32" id="label"> </text>

      </g>

15    <g id="skinContextMenu_bottom" display="all">

      <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-width:1" />

      <line x1="150" y1="35" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />

      <line x1="0" y1="38" x2="150" y2="38" style="stroke:#000000;stroke-width:1"
20  />

      <rect x="1" y="36" width="215" height="2" style="fill:#FFFFFF"/>

    </g>

  </g>

</svg>

```

25

### The 'comboBox' Skin Template

```

<?xml version="1.0"?>

<svg>

  <g id="skin">

    <use id="TextBoxSkin" xlink:href="" x="" y="" />

5    <use id="ButtonSkin" xlink:href="" x="" y="" />

    <use id="DropDownSkin" xlink:href="" x="" y="" />

    <g id="label">

      <text id="skin_bodyText" x="" y=""> </text>

    </g>

10  </g>

  <g id="skinDropDown">

    <g id="skinDropDown_top">

      </g>

15    <g id="skinContextMenu_middle" display="all">

      <text x="" y="" id="label"> </text>

      </g>

      <g id="skinContextMenu_middle_hover" display="none">

        <text x="" y="" id="label"> </text>

20    </g>

    <g id="skinDropDown_bottom">

      </g>

    </g>

25  </svg>

```

**Example 'comb Box' Skin**

```

<?xml version="1.0"?>

<svg>
5   <g id="skinListBox">
        <use id="TextBoxSkin" xlink:href="skinTextBox_ListBox.svg#skinTextbox" x="0"
y="0" />
        <use id="ButtonSkin" xlink:href="skinButton_ListBox.svg#skinButton" x="200"
y="0" />
10    <use id="DropDownSkin"
xlink:href="skinContextMenu_Default.svg#skinContextMenu" x="0" y="0" />
        <g id="label">
                <text id="skinListBoxDefault_bodyText" x="0" y="-6" fill="black"> </text>
        </g>
15    </g>

        <g id="skinDropDown">
                <g id="skinDropDown_top">
                        <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-width:1"
20    />
                        <line x1="217" y1="17" x2="217" y2="20"
style="stroke:#000000;stroke-width:1" />
                        <rect x="1" y="18" width="215" height="2" style="fill:#FFFFFF"/>
                </g>
25    <g id="skinContextMenu_middle" display="all">

```

```

        <line x1="0" y1="20" x2="0" y2="35"
style="stroke:#000000;stroke-width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />
5        <rect x="1" y="20" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="32" id="label"> </text>

    </g>

    <g id="skinContextMenu_middle_hover" display="none">
10        <line x1="0" y1="20" x2="0" y2="35"
style="stroke:#000000;stroke-width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
15 style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="32" id="label"> </text>

    </g>

    <g id="skinDropDown_bottom">
20        <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-width:1"
/>

        <line x1="217" y1="35" x2="217" y2="38"
style="stroke:#000000;stroke-width:1" />

        <line x1="0" y1="38" x2="217" y2="38"
25 style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="36" width="215" height="2" style="fill:#FFFFFF"/>

    </g>

```

```

    </g>
</svg>

```

### The 'scrollBar' Skin Template

```

5  <?xml version="1.0"?>

    <svg>

        <g id="skinScrollBar">

            <use id="SliderSkin" xlink:href="" x="" y=""/>

            <use id="ButtonLeftSkin" xlink:href="" x="" y=""/>

10         <use id="ButtonRightSkin" xlink:href="" x="" y=""/>

            <g id="corner">

                </g>

            </g>

            <g id="ButtonLeftSkin">

15         <g id="up" display="all">

                </g>

                <g id="down" display="none">

                </g>

            </g>

20         <g id="SliderSkin">

            <g id="up" display="all">

                <g id="body">

                    <g id="SliderSkinBar">

                    </g>

25         </g>

```



```

    <g id="SliderSkinThumb">
    </g>

    <pattern id="barfill" width="" height="" patternUnits="userSpaceOnUse">
    </pattern>
5    </g>

    <g id="mask" >
        <g id="sliderMask" >
        </g>
        </g>
10    </g>

    <g id="ButtonRightSkin">
        <g id="up" display="all">
        </g>
        <g id="down" display="none">
15    </g>
        </g>

    <g id="skinScrollBarSmall">
        <use id="SliderSkin" xlink:href="" x="" y=""/>
20    <use id="ButtonLeftSkin" xlink:href="" x="" y=""/>
        <use id="ButtonRightSkin" xlink:href="" x="" y=""/>
        <g id="corner">
        </g>
        </g>
25    <g id="ButtonLeftSkinSmall">

```

```

    <g id="up" display="all">
    </g>
    <g id="down" display="none">
    </g>
5    </g>
    <g id="SliderSkinSmall">
        <g id="up" display="all">
            <g id="SkinSliderSmallBorder" display="all">
            </g>
10        <g id="body">
            <g id="SkinSliderSmallBase">
            </g>
            <g id="SkinSliderSmallBar">
            </g>
15        </g>
            <g id="SliderSkinSmallThumb">
                <g id="sliderThumbBar">
                </g>
            </g>
20        <g id="tickMajor" display="none">
            <text x="" y="" id="tickLabelMajor" xml:space="preserve" style=""></text>
            </g>
        </g>
        <g id="mask" >
25        <g id="sliderMask" opacity="0">

```

```

    </g>
  </g>
</g>
<g id="ButtonRightSkinSmall">
5   <g id="up" display="all">
    </g>
    <g id="down" display="none">
    </g>
  </g>
10 </svg>

```

### Example 'scrollBar' Skin

```

<?xml version="1.0"?>
<svg>
15 <g id="skinScrollBar">
    <use id="SliderSkin" xlink:href="skinScrollBar_Default.svg#SliderSkin" x="0"
    y="0"/>
    <use id="ButtonLeftSkin" xlink:href="skinScrollBar_Default.svg#ButtonLeftSkin"
    x="0" y="0"/>
20 <use id="ButtonRightSkin"
    xlink:href="skinScrollBar_Default.svg#ButtonRightSkin" x="0" y="0"/>
    <g id="corner">
        <rect fill="rgb(212 208 200)" x="0" y="0" width="16" height="16" />
    </g>
25 </g>

```

```

<g id="ButtonLeftSkin">
  <g id="up" fill="none" stroke-linecap="square" shape-rendering="optimizeSpeed"
display="all">
    <rect fill="rgb(212 208 200)" width="16" height="16" />
5    <line x1="15" y1="0" x2="15" y2="15" style="stroke:#404040"/>
    <line x1="0" y1="15" x2="15" y2="15" style="stroke:#404040"/>
    <line x1="1" y1="1" x2="13" y2="1" style="stroke:#FFFFFF"/>
    <line x1="1" y1="1" x2="1" y2="13" style="stroke:#FFFFFF"/>
    <line x1="1" y1="14" x2="14" y2="14" style="stroke:#808080"/>
10    <line x1="14" y1="1" x2="14" y2="14" style="stroke:#808080"/>
    <path fill="black" d="M9.5 5.5l-3 3l3 3z" />
  </g>
  <g id="down" fill="none" stroke-linecap="square"
shape-rendering="optimizeSpeed" display="none">
15    <rect fill="rgb(212 208 200)" width="16" height="16" />
    <rect fill="none" style="stroke:#808080" width="16" height="16"/>
    <path fill="black" d="M10.5 6.5l-3 3l3 3z" />
  </g>
</g>
20 <g id="SliderSkin">
  <g id="up" display="all">
    <g id="body">
      <rect id="SliderSkinBar" fill="#E6E6E6" x="0" y="0" width="400"
height="16">
25    </rect>

```

```

    </g>

    <g id="SliderSkinThumb">
        <rect id="sliderThumbBar" style="fill:rgb(212 208
200);stroke:#000000;stroke-width:0" x="0" y="0" width="400" height="16"/>
5        <line x1="0" x2="400" y1="15" y2="15" style="stroke:#404040"/>
        <line x1="1" x2="398" y1="1" y2="1" style="stroke:#FFFFFF"/>
    </g>

    <pattern id="barfill" width="2" height="2" patternUnits="userSpaceOnUse"
shape-rendering="optimizeSpeed">
10    <rect x="0" y="0" width="2" height="2" fill="white"/>
        <rect x="0" y="0" width="1" height="1" fill="rgb(212 208 200)/>
        <rect x="1" y="1" width="1" height="1" fill="rgb(212 208 200)/>
    </pattern>

15    </g>
    <g id="mask" >
        <rect id="sliderMask" x="0" y="0" width="400" height="16" opacity="0"/>
    </g>
    </g>
20    <g id="ButtonRightSkin">
        <g id="up" fill="none" stroke-linecap="square" shape-rendering="optimizeSpeed"
display="all">
            <rect fill="rgb(212 208 200)" width="16" height="16" />
            <line x1="15" y1="0" x2="15" y2="15" style="stroke:#404040"/>
25    <line x1="0" y1="15" x2="15" y2="15" style="stroke:#404040"/>

```

```

    <line x1="1" y1="1" x2="13" y2="1" style="stroke:#FFFFFF"/>
    <line x1="1" y1="1" x2="1" y2="13" style="stroke:#FFFFFF"/>
    <line x1="1" y1="14" x2="14" y2="14" style="stroke:#808080"/>
    <line x1="14" y1="1" x2="14" y2="14" style="stroke:#808080"/>
5    <path fill="black" d="M6.5 5.5l3 3l-3 3z" />
    </g>
    <g id="down" fill="none" stroke-linecap="square"
shape-rendering="optimizeSpeed" display="none">
        <rect fill="rgb(212 208 200)" width="16" height="16" />
10    <rect fill="none" style="stroke:#808080" width="16" height="16"/>
        <path fill="black" d="M7.5 6.5l3 3l-3 3z" />
    </g>
</g>

15
<g id="skinScrollBarSmall">
    <use id="SliderSkin" xlink:href="skinScrollBar_Default.svg#SliderSkinSmall"
x="0" y="0"/>
    <use id="ButtonLeftSkin"
20 xlink:href="skinScrollBar_Default.svg#ButtonLeftSkinSmall" x="0" y="0"/>
    <use id="ButtonRightSkin"
xlink:href="skinScrollBar_Default.svg#ButtonRightSkinSmall" x="0" y="0"/>
    <g id="corner">
        <rect x="0" y="0" width="10" height="10"
25 style="fill:#FFFFFF;stroke:#000000"/>
    </g>

```

```

    </g>

    <g id="ButtonLeftSkinSmall">

        <g id="up" display="all">

            <rect x="1" y="1" width="10" height="10"
5      style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
            0001;stroke-miterlimit:2.6131;"/>

            <rect x="0" y="0" width="10" height="10"
            style="fill:#FFFFFF;stroke:#000000"/>

            <text x="5" y="5"
10      style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

            <path d="M 8 2 1 -6 3 1 6 3 z"
            style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

            </g>

            <g id="down" display="none">

15          <rect x="1" y="1" width="10" height="10"
            style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
            0001;stroke-miterlimit:2.6131;"/>

            <rect x="1" y="1" width="10" height="10"
            style="fill:#FFFFFF;stroke:#000000"/>

20          <text x="5" y="5"
            style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

            <path d="M 9 3 1 -6 3 1 6 3 z"
            style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

            </g>

25      </g>

      <g id="SliderSkinSmall">

        <g id="up" display="all">

```

```

    <rect id="SkinSliderSmallBorder" x="0" y="0" width="200" height="10"
opacity="0"/>

    <g id="body">

        <rect id="SkinSliderSmallBase"
5   style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="1" y="1" width="444" height="10"/>

        <rect id="SliderSkinSmallBar" style="fill:#FFFFFF;stroke:#000000" x="0"
y="0" width="444" height="10"/>

        </g>

10    <g id="SliderSkinSmallThumb">

        <rect id="sliderThumbBar"
style="fill:#B3B3B3;stroke:#000000;stroke-width:0" x="0" y="0" width="10"
height="10"/>

        </g>

15    <g id="tickMajor" display="none">

        <text x="0" y="11" id="tickLabelMajor" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:black;stroke:1;text-anchor:middle"></text>

        </g>

        </g>

20    <g id="mask">

        <rect id="sliderMask" x="" y="0" width="444" height="18" opacity="0"/>

        </g>

        </g>

        <g id="ButtonRightSkinSmall">

25    <g id="up" display="all">

```



```

    <rect x="1" y="1" width="10" height="10"
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;"/>

    <rect x="0" y="0" width="10" height="10"
5 style="fill:#FFFFFF;stroke:#000000"/>

    <text x="5" y="5"
style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

    <path d="M 2 2 1 6 3 1 -6 3 z"
style="fill:#000000;stroke:#FF0000;stroke-width:0"/>
10 </g>

    <g id="down" display="none">

        <rect x="1" y="1" width="10" height="10"
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;"/>

15 <rect x="1" y="1" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

        <text x="5" y="5"
style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

        <path d="M 3 3 1 6 3 1 -6 3 z"
20 style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

    </g>

    </g>

</svg>

25 The 'slider' Skin Template

<?xml version="1.0"?>

<svg>

```

```

5  <g id="skinSlider">
    <g id="label">
        <text x="" y="" style=""> </text>
    </g>

10  <g id="up" display="all">
    <g id="skinSliderBorder">
        </g>
    <g id="body">
        <g id="skinSliderBase">
            </g>
            <g id="skinSliderBar">
                </g>
            </g>
15  <g id="skinSliderThumb">
            </g>
            <g id="tickMajor" display="none">
                <text id="tickLabelMajor" x="" y="" xml:space="preserve" style=""></text>
            </g>
20  <g id="tickMinor" display="none">
            </g>
        </g>

25  <g id="down" display="none">
    <g id="skinSliderBorder">

```

```

    </g>

    <g id="body">
        <g id="skinSliderBase">
            </g>
5      <g id="skinSliderBar">
            </g>
        </g>
        <g id="skinSliderThumb">
            </g>
10     <g id="tickMajor" display="none">
            <text id="tickLabelMajor" x="" y="" xml:space="preserve" style=""></text>
            </g>
            <g id="tickMinor" display="none">
                </g>
15     </g>

    <g id="hover" display="none">
        <g id="skinSliderBorder">
            </g>
20     <g id="body">
        <g id="skinSliderBase">
            </g>
            <g id="skinSliderBar">
                </g>
25     </g>

```

```

    <g id="skinSliderThumb">
    </g>
    <g id="tickMajor" display="none">
        <text id="tickLabelMajor" x="" y="" xml:space="preserve" style=""></text>
5    </g>
    <g id="tickMinor" display="none">
    </g>
</g>

10
<g id="disabled" display="none">
    <g id="skinSliderBorder">
    </g>
    <g id="body">
15    <g id="skinSliderBase">
    </g>
    <g id="skinSliderBar">
    </g>
    </g>
20 <g id="skinSliderThumb">
    </g>
    <g id="tickMajor" display="none">
        <text id="tickLabelMajor" x="" y="" xml:space="preserve" style=""></text>
    </g>
25 <g id="tickMinor" display="none">

```

```
</g>
```

```
</g>
```

```
<g id="mask" >
```

```
5      <g id="skinSliderMask">
```

```
      </g>
```

```
    </g>
```

```
  </g>
```

```
</svg>
```

```
10
```

### Example 'slider' Skin

```
<?xml version="1.0"?>
```

```
<svg>
```

```
  <g id="skinSlider">
```

```
15    <g id="label">
```

```
      <text x="210" y="18" style="font-family:'Tahoma';stroke:none;font-size:12;">
```

```
    </text>
```

```
  </g>
```

```
  <g id="up" display="all">
```

```
20    <rect id="skinSliderBorder" x="0" y="0" width="200" height="18" fill="none"
stroke="none" pointer-events="fill"/>
```

```
    <g id="body">
```

```
      <rect id="skinSliderBase" style="fill:#FFFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>
```

```
25    <rect id="skinSliderBar" style="fill:#FFFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>
```

```

    <line x1="0" y1="7" x2="200" y2="7" style="stroke:#B3B3B3"/>
  </g>
  <g id="skinSliderThumb">
    <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
5    <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>
    <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3" width="6"
height="12"/>
    <line x1="0" y1="3" x2="0" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>
    <line x1="1.5" y1="3" x2="1.5" y2="15" style="stroke:white;stroke-width:2"/>
10    <line x1="6" y1="3" x2="6" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>
  </g>
  <g id="tickMajor" display="none">
    <line x1="0" y1="7" x2="0" y2="17" style="stroke:#B3B3B3"/>
    <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
15 style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke:1;text-anchor:middle"></text>
  </g>
  <g id="tickMinor" display="none">
    <line x1="0" y1="7" x2="0" y2="11" style="stroke:#B3B3B3"/>
  </g>
20 </g>

  <g id="down" display="none">
    <rect id="skinSliderBorder" x="0" y="0" width="200" height="18" fill="none"
stroke="none" pointer-events="fill"/>
25 <g id="body">

```

```

    <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>

    <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>
5      <line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>

    </g>

    <g id="skinSliderThumb">

        <circle cx="3" cy="3" r="3" style="fill:#199DBF;stroke:none"/>

        <circle cx="3" cy="15" r="3" style="fill:#199DBF;stroke:none"/>
10      <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3" width="6"
height="12"/>

        <line x1="0" y1="3" x2="0" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>

        <line x1="1.5" y1="3" x2="1.5" y2="15" style="stroke:white;stroke-width:2"/>

        <line x1="6" y1="3" x2="6" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>
15    </g>

    <g id="tickMajor" display="none">

        <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>

        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:black;stroke:1;text-anchor:middle"></text>
20    </g>

    <g id="tickMinor" display="none">

        <line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>

    </g>

    </g>
25

    <g id="hover" display="none">

```

```

    <rect id="skinSliderBorder" x="0" y="0" width="200" height="18" fill="none"
stroke="none" pointer-events="fill"/>

    <g id="body">

        <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none" x="0" y="7"
5 width="200" height="4"/>

        <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>

        <line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>

    </g>

10 <g id="skinSliderThumb">

    <circle cx="3" cy="3" r="3" style="fill:#199DBF;stroke:none"/>

    <circle cx="3" cy="15" r="3" style="fill:#199DBF;stroke:none"/>

    <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3" width="6"
height="12"/>

15 <line x1="0" y1="3" x2="0" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>

    <line x1="1.5" y1="3" x2="1.5" y2="15" style="stroke:white;stroke-width:2"/>

    <line x1="6" y1="3" x2="6" y2="15" style="stroke:#B3B3B3;stroke-width:1"/>

    </g>

    <g id="tickMajor" display="none">

20 <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>

    <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:black;stroke:1;text-anchor:middle"></text>

    </g>

    <g id="tickMinor" display="none">

25 <line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>

    </g>

```



```

</g>

<g id="disabled" display="none">
  <rect id="skinSliderBorder" x="0" y="0" width="200" height="18" fill="none"
5  stroke="none" pointer-events="fill"/>
  <g id="body">
    <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none" x="0" y="7"
width="200" height="4"/>
    <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none" x="0" y="7"
10 width="200" height="4"/>
    <line x1="0" y1="7" x2="200" y2="7" style="stroke:#B3B3B3"/>
  </g>
  <g id="skinSliderThumb">
    <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
    <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>
15  <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3" width="6"
height="12"/>
    <line x1="0" y1="3" x2="0" y2="15" style="#stroke:B3B3B3;stroke-width:1"/>
    <line x1="1.5" y1="3" x2="1.5" y2="15" style="stroke:white;stroke-width:2"/>
    <line x1="6" y1="3" x2="6" y2="15" style="#stroke:000000;stroke-width:1"/>
20  </g>
  <g id="tickMajor" display="none">
    <line x1="0" y1="7" x2="0" y2="17" style="stroke:#B3B3B3"/>
    <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
25 style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke:1;text-anchor:middle"></text>
  </g>

```

```
<g id="tickMinor" display="none">
```

```
<line x1="0" y1="7" x2="0" y2="11" style="stroke:#B3B3B3"/>
```

```
</g>
```

```
</g>
```

5

```
<g id="mask" >
```

```
<rect id="skinSliderMask" x="-4" y="0" width="208" height="18" fill="none"
stroke="none" pointer-events="fill"/>
```

```
</g>
```

10 &lt;/g&gt;

```
</svg>
```

### The 'spin' Skin Template

```
<?xml version="1.0"?>
```

15 &lt;svg&gt;

```
<g id="skinSpin">
```

```
<use id="TextBoxSkin" xlink:href="" x="" y="" />
```

```
<use id="ButtonSkinUp" xlink:href="" x="" y="" />
```

```
<use id="ButtonSkinDown" xlink:href="" x="" y="" />
```

20

```
<g id="label">
```

```
<text x="" y="" style=""> </text>
```

```
</g>
```

```
</g>
```

25

```

    <g id="skinSpinButtonUp">
      <g id="up" display="all">
        </g>
      <g id="down" display="none">
5      </g>
      <g id="hover" display="none">
        </g>
      <g id="disabled" transform="scale(0.5)" display="none">
        </g>
10    </g>

```

```

    <g id="skinSpinButtonDown">
      <g id="up" display="all">
        </g>
15    <g id="down" display="none">
        </g>
      <g id="hover" display="none">
        </g>
      <g id="disabled" display="none">
20    </g>
    </g>
  </svg>

```

### Example 'spin' Skin

```

25 <?xml version="1.0"?>

```

```

<svg>
  <g id="skinSpin">
    <use id="TextBoxSkin" xlink:href="skinTextBox_Listbox_Short.svg#skinTextbox"
x="0" y="0"/>
5    <use id="ButtonSkinUp" xlink:href="skinSpin_Composite.svg#skinSpinButtonUp"
x="100" y="-1"/>
    <use id="ButtonSkinDown"
xlink:href="skinSpin_Composite.svg#skinSpinButtonDown" x="100" y="8"/>
    <g id="label">
10    <text x="0" y="-6" style="font-size:12;fill:black"> </text>
    </g>
  </g>

  <g id="skinSpinButtonUp">
15  <g id="up" display="all">
    <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>
    <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:white;fill:none"/>
    <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:black;fill:none"/>
20    <polygon points="3,6 9,3 15,6" style="stroke-width:1;stroke:black;fill:black"/>
    </g>
    <g id="down" display="none">
    <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>
25    <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:black;fill:none"/>
    <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:white;fill:none"/>

```

```

        <polygon points="3,6 9,3 15,6"
style="stroke-width:1;stroke:#199DBF;fill:#199DBF"/>
    </g>
    <g id="hover" display="none">
5        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,9 0,0 18,0" style="stroke-width:2;stroke:white;fill:none"/>
        <polyline points="0,9 18,9 18,0" style="stroke-width:2;stroke:black;fill:none"/>
        <polygon points="3,6 9,3 15,6"
10 style="stroke-width:1;stroke:#199DBF;fill:#199DBF"/>
    </g>
    <g id="disabled" style="opacity:0.5" transform="scale(0.5)" display="none">
        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>
15        <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:white;fill:none"/>
        <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:black;fill:none"/>
        <polygon points="3,6 9,3 15,6" style="stroke-width:1;stroke:black;fill:black"/>
    </g>
</g>
20
<g id="skinSpinButtonDown">
    <g id="up" display="all">
        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>
25        <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:white;fill:none"/>
        <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:black;fill:none"/>

```

```

    <polygon points="3,3 9,6 15,3" style="stroke-width:1;stroke:black;fill:black"/>
  </g>

  <g id="down" display="none">
    <rect x="0" y="0" width="18" height="9"
5    style="fill:rgb(230,230,230);stroke:none"/>
    <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:black;fill:none"/>
    <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:white;fill:none"/>
    <polygon points="3,3 9,6 15,3"
    style="stroke-width:1;stroke:#199DBF;fill:#199DBF"/>
10  </g>

    <g id="hover" display="none">
      <rect x="0" y="0" width="18" height="9"
      style="fill:rgb(230,230,230);stroke:none"/>
      <polyline points="0,9 0,0 18,0" style="stroke-width:2;stroke:white;fill:none"/>
15  <polyline points="0,9 18,9 18,0" style="stroke-width:2;stroke:black;fill:none"/>
      <polygon points="3,3 9,6 15,3"
      style="stroke-width:1;stroke:#199DBF;fill:#199DBF"/>
      </g>

      <g id="disabled" style="opacity:0.5" display="none">
20  <rect x="0" y="0" width="18" height="9"
      style="fill:rgb(230,230,230);stroke:none"/>
      <polyline points="0,9 0,0 18,0" style="stroke-width:1;stroke:white;fill:none"/>
      <polyline points="0,9 18,9 18,0" style="stroke-width:1;stroke:black;fill:none"/>
      <polygon points="3,3 9,6 15,3" style="stroke-width:1;stroke:black;fill:black"/>
25  </g>
    </g>

```

```
</svg>
```

### The 'textBox' Skin Template

```
<?xml version="1.0"?>
```

```
5 <svg>
```

```
    <g id="skinTextboxScroll">
```

```
        <use id="TextBoxSkin" xlink:href="" x="#skinTextbox" y=""/>
```

```
        <use id="VertScrollSkin" xlink:href="" x="" y=""/>
```

```
        <use id="HorizScrollSkin" xlink:href="" x="" y=""/>
```

```
10    </g>
```

```
    <g id="skinTextbox">
```

```
        <g id="up" display="all">
```

```
        </g>
```

```
15    <g id="hover" display="none">
```

```
    </g>
```

```
    <g id="down" display="none">
```

```
    </g>
```

```
    <g id="disabled" display="none">
```

```
20    </g>
```

```
    <g id="label">
```

```
        <text x="238" y="14" style="font-size:12;fill:black"> </text>
```

```
    </g>
```

```
    <clipPath id="skinTextbox_textView">
```

```
25    </clipPath>
```

```

    <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="" y="" style=""></text>

  </g>

</svg>

```

5

### Example 'textBox' Skin

```

<?xml version="1.0"?>

<svg>

  <g id="skinTextboxScroll">
10    <use id="TextBoxSkin" xlink:href="#skinTextbox" x="0" y="0"/>

    <use id="VertScrollSkin"
xlink:href="skinScrollBar_Default.svg#skinScrollBarSmall" x="0" y="0"/>

    <use id="HorizScrollSkin"
xlink:href="skinScrollBar_Default.svg#skinScrollBarSmall" x="0" y="0"/>
15  </g>

  <g id="skinTextbox">

    <g id="up"
style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;"
20  display="all">

      <rect
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="18"/>

      <rect style="fill:white;stroke:grey" x="0" y="0" width="228" height="18"/>
25  </g>

```



```

    <g id="hover"
style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;"
display="none">

        <rect
5   style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="18"/>

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228" height="18"/>

    </g>

    <g id="down"
10  style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;"
display="none">

        <rect
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="18"/>

15  <rect style="fill:white;stroke:#000000" x="0" y="0" width="228" height="18"/>

        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black">

            <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

20  </line>

    </g>

    <g id="disabled"
style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;"
display="none">

25  <rect
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="18"/>

        <rect style="fill:gray;stroke:#000000" x="0" y="0" width="228" height="18"/>

```

```

    </g>

    <g id="label">
        <text x="238" y="14" style="font-size:12;fill:black"> </text>
    </g>
5    <clipPath id="skinTextbox_textView">
        <rect id="skinTextbox_textArea" x="3" y="3" width="222" height="12"/>
    </clipPath>

    <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
10    xml:space="preserve" x="3" y="15"
    style="font-face:Arial;font-size:12;fill:black;stroke:1"></text>
    </g>

    <g id="skinTextboxMultiLine">
        <g id="up"
15    style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;"
    display="all">
            <rect
                style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
                0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="54"/>
20    <rect style="fill:white;stroke:grey" x="0" y="0" width="228" height="54"/>
        </g>

        <g id="hover"
            style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;"
            display="none">
25    <rect
                style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
                0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="54"/>

```

```

    <rect style="fill:white;stroke:#000000" x="0" y="0" width="228" height="54"/>

    </g>

    <g id="down"
style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4,"
5  display="none">

        <rect
style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="54"/>

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228" height="54"/>
10    <line id="skinTextboxMultiLine_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black">

        <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

        </line>

15    </g>

    <g id="disabled"
style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;
" display="none">

        <rect
20 style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.
0001;stroke-miterlimit:2.6131;" x="3" y="3" width="228" height="54"/>

        <rect style="fill:gray;stroke:#000000" x="0" y="0" width="228" height="54"/>

        </g>

        <clipPath id="skinTextboxMultiLine_textView">

25    <rect id="skinTextboxMultiLine_textArea" fill="green" x="3" y="3" width="222"
height="48"/>

        </clipPath>

```

```

    <text clip-path="url(#skinTextboxMultiLine_textView)"
id="skinTextboxMultiLine_text" xml:space="preserve" x="3" y="15"
style="font-face:Arial;font-size:12;fill:black;stroke:none"/>
  </g>
5 </svg>

```

### The 'toolTip' Skin Template

```

<?xml version="1.0"?>

<svg>
10   <g id="skinTooltip_name">
      <text x="" y="" style="">tip</text>
      </g>
    </svg>

```

### 15 Example 'toolTip' Skin

```

<?xml version="1.0"?>

<svg>
  <g id="skinTooltip_annotation">
    <path style="fill:#FFFF00;fill-opacity:0.7;stroke:#1F1A17;stroke-width:3"
20   d="M0 96 198 0 -56 -96 106 96 202 0 0 167 -352 0 0 -167z"/>
    <text x="10" y="150" style="font-size:16;fill:black">tip</text>
    </g>

    <g id="skinTooltip_traditional" pointer-events="none">
25   <rect x="0" y="0" width="150" height="15"

```

```

        style="fill:rgb(255,255,225);fill-opacity:0.9" stroke="black" stroke-width="1"/>
        <text x="5" y="12" style="font-size:12;fill:black">tip</text>
    </g>

```

```

5    <g id="skinTooltip_special" pointer-events="none">
        <rect x="0" y="0" width="150" height="15"
            style="fill:rgb(130,130,130);fill-opacity:0.5" stroke="black" stroke-width="1"/>
        <text x="5" y="12" style="font-size:12;fill:black">tip</text>
    </g>

```

```

10 </svg>

```

#### Other Example Skins for UI Controls

##### **skinButton\_ComboBox.svg**

```

    <?xml version="1.0"?>
15 <svg xmlns:dsvg="http://corel.org/dsvg" >
    <g id="skinButton">
        <g id="label">
            <text x="50" y="13" style="font-family:'Tahoma';stroke:none;font-
            size:12;text-anchor:middle"> </text>
20    </g>
        <g id="up" display="all" style="stroke-width:1">
            <rect x="0" y="0" width="17" height="17"
            style="fill:#000000;stroke:#A3A3A3"/>
            <path d="M 5 5 16 0 13 6 z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
25    <dsvg:constraint preserveAspectRatio="vertical" hAlign="right"/>

```

```

    </g>
    <g id="down" display="none">
        <rect x="0" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#A3A3A3"/>
5        <path d="M 5 5 16 0 1-3 6 z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
        <svg:constraint preserveAspectRatio="vertical" hAlign="right"/>
    </g>
    <g id="hover" display="none">
        <rect x="0" y="0" width="17" height="17"
10 style="fill:#199DBF;stroke:#A3A3A3"/>
        <path d="M 5 5 16 0 1-3 6 z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
        <svg:constraint preserveAspectRatio="vertical" hAlign="right"/>
    </g>
</g>
15 </svg>

```

### skinButton\_ListBox.svg

```

<?xml version="1.0"?>
<svg>
20 <g id="skinListBox">
    <g id="up" display="all">
        <g id="skinListBox_label">
            <text x="0" y="-5" style="fill:#000000;font-
weight:normal;font-size:12;font-family:Tahoma"> </text>
25 </g>

```

```

    <g id="skinListBox_body" style="fill:none">
        <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
        <rect x="200" y="0" width="17" height="17"
5 style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>
        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>
        <text id="skinListBox_bodyText" x="5" y="13"
style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>
10 </g>
        <g id="skinListBox_bodyCover">
            <rect x="0" y="0" width="200" height="17" fill="none"
stroke="none" pointer-events="fill"/>
            </g>
15 <g id="skinListBox_dropDownButton">
            <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#000000;stroke-width:1"/>
            <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>
20 </g>
        </g>

        <g id="down" display="none">
            <g id="skinListBox_label">
25 <text x="0" y="-5" style="fill:#000000;font-
weight:normal;font-size:12;font-family:Tahoma"> </text>

```

```

    </g>

    <g id="skinListBox_body" style="fill:none">

        <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
5        <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>

        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

        <text id="skinListBox_bodyText" x="5" y="13"
10 style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>

    </g>

    <g id="skinListBox_bodyCover">

        <rect x="0" y="0" width="200" height="17" fill="none"
stroke="none" pointer-events="fill"/>

15    </g>

    <g id="skinListBox_dropDownButton">

        <rect x="200" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#199DBF;stroke-width:1"/>

        <path d="M 205 5 1 6 0 1 -3 6 z"
20 style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

    </g>

</g>

<g id="hover" display="none">

25    <g id="skinListBox_label">

```



```

        <text x="0" y="-5" style="fill:#000000;font-
weight:normal;font-size:12;font-family:Tahoma"> </text>

```

```

    </g>

```

```

    <g id="skinListBox_body" style="fill:none">

```

```

5      <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

```

```

        <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>

```

```

10      <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

```

```

        <text id="skinListBox_bodyText" x="5" y="13"
style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>

```

```

    </g>

```

```

    <g id="skinListBox_bodyCover">

```

```

15      <rect fill="none" stroke="none" pointer-events="fill" x="0"
y="0" width="200" height="17"/>

```

```

    </g>

```

```

    <g id="skinListBox_dropDownButton">

```

```

20      <rect x="200" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#199DBF;stroke-width:1"/>

```

```

        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

```

```

    </g>

```

```

  </g>

```

```

25

```

```

    <g id="skinListBox_dropDownList">

```

```

    <g id="skinListBox_dropDownList_top">
        <line x1="0" y1="17" x2="0" y2="20"
5 style="stroke:#000000;stroke-width:1" />
        <line x1="217" y1="17" x2="217" y2="20"
5 style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="18" width="215" height="2"
style="fill:#FFFFFF"/>
    </g>

    <g id="skinListBox_dropDownList_middle">
10 <line x1="0" y1="20" x2="0" y2="35"
style="stroke:#000000;stroke-width:1" />
        <line x1="217" y1="20" x2="217" y2="35"
style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="20" width="215" height="14"
15 style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>
    </g>

    <g id="skinListBox_dropDownList_bottom">
        <line x1="0" y1="35" x2="0" y2="38"
style="stroke:#000000;stroke-width:1" />
20 <line x1="217" y1="35" x2="217" y2="38"
style="stroke:#000000;stroke-width:1" />
        <line x1="0" y1="38" x2="217" y2="38"
style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="36" width="215" height="2"
25 style="fill:#FFFFFF"/>
    </g>
</g>

```

```
</g>
```

```
</svg>
```

### skinButton\_vpDown.svg

```
5 <?xml version="1.0"?>
```

```
<svg>
```

```
<g id="skinButton">
```

```
<g id="label">
```

```
10 <text x="50" y="13" style="font-family:'Tahoma';stroke:none;font-  
size:12;text-anchor:middle"> </text>
```

```
</g>
```

```
<g id="up" display="all">
```

```
<path fill="#CCCCCC" d="M4 115 11 5 -11 -10 0z"/>
```

```
<path fill="#E5FFFF" stroke="#0083C4" d="M1 115 11 5 -11 -10 0z"/>
```

```
15 </g>
```

```
<g id="down" display="none">
```

```
<path fill="#CCCCCC" d="M4 115 11 5 -11 -10 0z"/>
```

```
<path fill="#EDAF0D" stroke="#0083C4" stroke-width="2" d="M1 115 11 5 -11 -10  
0z"/>
```

```
20 </g>
```

```
<g id="hover" display="none">
```

```
<path fill="#CCCCCC" d="M4 115 11 5 -11 -10 0z"/>
```

```
<path fill="#E5FFFF" stroke="#0083C4" stroke-width="2" d="M1 115 11 5 -11 -10 0z"/>
```

```
</g>
```

```
25 <g id="disabled" style="opacity:0.5" display="none">
```

```
</g>
```

```
</g>
```

```
</svg>
```

## 5 skinButton\_vpGeneric.svg

```
<?xml version="1.0"?>
```

```
<svg>
```

```
<g id="skinButton">
```

```
<g id="label">
```

```
10      <text x="9" y="12" fill="#000000" font-weight="normal" font-
size="10" font-family="Verdana"> </text>
```

```
</g>
```

```
<g id="up" display="all">
```

```
<path fill="#CCCCCC" d="M56 181-52 0 0 -15 52 0 0 15z"/>
```

```
15 <path fill="#FFFFFF" stroke="#0083C4" d="M54 151-53 0 0 -15 53 0 0 15z"/>
```

```
</g>
```

```
<g id="down" display="none">
```

```
<path fill="#CCCCCC" d="M56 181-52 0 0 -15 52 0 0 15z"/>
```

```
<path fill="#EDAF0D" stroke="#0083C4" stroke-width="2" d="M54 151-53 0 0 -15 53
20 0 0 15z"/>
```

```
</g>
```

```
<g id="hover" display="none">
```

```
<path fill="#CCCCCC" d="M56 181-52 0 0 -15 52 0 0 15z"/>
```

```
<path fill="#FFFFFF" stroke="#0083C4" stroke-width="2" d="M54 151-53 0 0 -15 53
25 0 0 15z"/>
```

```

    </g>

    <g id="disabled" style="opacity:0.5" display="none">

    <path fill="#CCCCCC" d="M56 181-52 0 0 -15 52 0 0 15z"/>

    <path fill="#FFFFFF" stroke="#CCCCCC" d="M54 151-53 0 0 -15 53 0 0 15z"/>
5
    </g>

    </g>

</svg>

```

# 10 skinButton\_vpMonth.svg

```

<?xml version="1.0"?>

<svg>

    <g id="skinButton">

        <g id="label">

15            <text x="22" y="11" fill="#666666" font-weight="bold" font-size="7"
font-family="Verdana"> </text>

            </g>

            <g id="up" display="all">

                <rect fill="#B2E5E5" x="0" y="14" width="40" height="4"/>

20            <rect fill="#B2E5E5" x="0" y="0" width="40" height="4"/>

            <rect fill="#FFFFFF" fill-opacity="0" x="0" y="0" width="40" height="18"/>

            </g>

            <g id="down" display="none">

                <rect fill="#98B2E5" x="0" y="14" width="40" height="4"/>

25            <rect fill="#98B2E5" x="0" y="0" width="40" height="4"/>

```

```

<rect fill="#FFFFFF" fill-opacity="0" x="0" y="0" width="40" height="18"/>
    </g>
    <g id="hover" display="none">
        <rect fill="#98B2E5" x="0" y="14" width="40" height="4"/>
5      <rect fill="#98B2E5" x="0" y="0" width="40" height="4"/>
        <rect fill="#FFFFFF" fill-opacity="0" x="0" y="0" width="40" height="18"/>
            </g>
            <g id="disabled" display="none">
                <rect fill="#CCCCCC" x="0" y="14" width="40" height="4"/>
10      <rect fill="#CCCCCC" x="0" y="0" width="40" height="4"/>
                <rect fill="#FFFFFF" fill-opacity="0" x="0" y="0" width="40" height="18"/>
                    </g>
                </g>
            </svg>
15
skinButton_vpUp.svg
<?xml version="1.0"?>
<svg>
    <g id="skinButton">
20      <g id="label">
            <text x="50" y="13" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>
            </g>
            <g id="up" display="all">
25      <path fill="#CCCCCC" d="M15 14l-4 -11 -5 11 9 0z"/>

```

```

    <path fill="#E5FFFF" stroke="#0083C4" d="M12 131-5 -12 -5 12 10 0z"/>
    </g>
    <g id="down" display="none">
    <path fill="#CCCCCC" d="M15 141-4 -11 -5 11 9 0z"/>
5    <path fill="#EDAF0D" stroke="#0083C4" stroke-width="2" d="M12 131-5 -12 -5 12 10
    0z"/>
    </g>
    <g id="hover" display="none">
    <path fill="#CCCCCC" d="M15 141-4 -11 -5 11 9 0z"/>
10    <path fill="#E5FFFF" stroke="#0083C4" stroke-width="2" d="M12 131-5 -12 -5 12 10
    0z"/>
    </g>
    <g id="disabled" style="opacity:0.5" display="none">
    </g>
15    </g>
    </svg>

```

# skinButton\_Windows.svg

```

<?xml version="1.0"?>
20 <svg xmlns:dsvg="http://corel.org/dsvg">
    <g id="skinButton">
        <g id="label" style="font-family:'Tahoma';stroke:none;font-size:12;text-
        anchor:middle">
            <text x="50" y="13"> </text>
25    <dsvg:constraint scaleImmunity="true" hAlign="middle"
        vAlign="middle"/>

```

```

    </g>

    <g id="up" display="all" style="stroke-width:1">
        <rect x="0" y="0" width="100" height="18"
5      style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0" style="stroke:white;fill:none"/>
        <polyline points="0,18 100,18 100,0" style="stroke:black;fill:none"/>
        <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
    </g>

10    <g id="down" display="none" style="stroke-width:1">
        <rect x="0" y="0" width="100" height="18"
        style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0" style="stroke:black;fill:none"/>
        <polyline points="0,18 100,18 100,0" style="stroke:white;fill:none"/>
15    <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
    </g>

    <g id="hover" display="none" style="stroke-width:2">
        <rect x="0" y="0" width="100" height="18"
20    style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0" style="stroke:white;fill:none"/>
        <polyline points="0,18 100,18 100,0" style="stroke:black;fill:none"/>
        <rect x="1" y="1" width="98" height="16"
        style="fill:none;stroke:#199DBF;stroke-width:1">
25    <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>

```



```

        </rect>

        <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
    </g>
5    <g id="disabled" display="none" style="stroke-width:1;opacity:0.5">
        <rect x="0" y="0" width="100" height="18"
style="fill:rgb(230,230,230);stroke:none"/>
        <polyline points="0,18 0,0 100,0" style="stroke:white;fill:none"/>
        <polyline points="0,18 100,18 100,0" style="stroke:black;fill:none"/>
10    <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
    </g>
</g>
</svg>
15
skinCalendar.svg
<?xml version="1.0"?>
<svg>
    <g id="skinCalendar">
20        <use id="CalendarBackSkin"
xlink:href="skinCalendar.svg#skinCalendarBack" x="0" y="0"/>
        <use id="DaysButtonsSkin" xlink:href="skinCalendar.svg#skinDaysButtons"
x="1" y="40"/>
    </g>
25    <g id="skinCalendarBack">
        <g id="up">

```

```

<g id="calendarBack">
  <path fill="#5D80D5" d="M243 15l0 -14 -131 0 -18 13 149 0z"/>
  <path fill="#F2F2F2" d="M244 29l-244 0 0 -3 244 0 0 3z"/>
  <path fill="#F2F2F2" d="M244 23l-244 0 0 -3 244 0 0 3z"/>
5   <path fill="#F2F2F2" d="M244 34l-244 0 0 -3 244 0 0 3z"/>
    <text x="15" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">S</text>
    <text x="48" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">M</text>
10   <text x="85" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">T</text>
    <text x="119" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">W</text>
    <text x="157" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
15 family="Verdana">T</text>
    <text x="190" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">F</text>
    <text x="224" y="30" fill="#5D80D5" font-weight="normal" font-size="9" font-
family="Verdana">S</text>
20   <path fill="none" stroke="#666666" d="M0 18l244 0"/>
    <!--<rect fill="none" x="9" y="8" width="287" height="287"/>-->
  </g>
  </g>
  <g id="down">
25   <use id="backDown" xlink:href="#calendarBack" x="0" y="0"/>
  </g>

```

```

    <g id="hover">
    <use id="backHover" xlink:href="#calendarBack" x="0" y="0"/>
    </g>
    <g id="disabled">
5    <use id="backDisabled" xlink:href="#calendarBack" x="0" y="0"/>
    </g>
    <g id="label">
    <text text-anchor="end" x="240" y="12" fill="#FFFFFF" font-weight="normal" font-
size="11" font-family="Verdana"> </text>
10    </g>
    </g>

    <g id="skinDaysButtons">
    <g id="up">
15    <rect height="33" width="35" stroke="none" fill="none"/>
    <path fill="#E5FFFF" stroke="#0083C4" d="M31 291-31 0 0 -29 31
0 0 29z"/>
    </g>
    <g id="down">
20    <path fill="#EDAF0D" stroke="#0083C4" d="M31 291-31 0 0 -29 31
0 0 29z"/>
    </g>
    <g id="hover">
    <path fill="#E5FFFF" stroke="#0083C4" stroke-width="2" d="M31
25 291-31 0 0 -29 31 0 0 29z"/>
    </g>

```

```

    <g id="focusUp">
        <path fill="#E5FFFF" stroke="#0083C4" d="M31 29l-31 0 0 -29 31
0 0 29z"/>
    </g>
5    <g id="disabledUp">
        <path fill="#F7FFFF" stroke="#98B2E5" d="M31 29l-31 0 0 -29 31
0 0 29z"/>
    </g>
    <g id="label">
10    <text text-anchor="middle" x="20" y="23" fill="#666666" font-weight="normal" font-
size="10" font-family="Verdana"> </text>
    </g>
    </g>
    <g id="skinDaysButtonsNonPaid">
15    <g id="down">
        <path fill="green" stroke="#0083C4" d="M31 29l-31 0 0 -29 31 0 0
29z"/>
    </g>
    </g>
20    <g id="skinDaysButtonsHalfDayNonPaid">
        <g id="down">
            <path fill="blue" stroke="#0083C4" d="M31 29l-31 0 0 -29 31 0 0
29z"/>
        </g>
    </g>
25    </g>
    <g id="skinDaysButtonsHalfDayPaid">

```

```

    <g id="down">
        <path fill="red" stroke="#0083C4" d="M31 291-31 0 0 -29 31 0 0
29z"/>
    </g>
5    </g>
    </svg>

skinCheckbox_Default.svg
<?xml version="1.0" standalone="no" ?>
10 <svg>
    <g id="skinCheckbox">
        <g id="label">
            <text x="15" y="10.5" style="font-weight:normal;font-size:12;font-
family:Tahoma; fill:#000000;"> </text>
15    </g>
        <g id="up" display="all">
            <rect x="0" y="0" width="12" height="12"
style="stroke:black;fill:white" />
        </g>
20    <g id="down" display="none">
            <rect x="0" y="0" width="12" height="12"
style="stroke:black;fill:white" />
            <path d="M 2 6 1 2 4 1 6.5 -8"
style="fill:#FFFFFF;stroke:#000000;stroke-width:1" />
25    </g>
        <g id="hover" display="none">

```

```

                <rect x="0" y="0" width="12" height="12"
style="stroke:black;fill:white" />

```

```

        </g>

```

```

        <g id="disabled" style="opacity:0.5" display="none">

```

```

5          <rect x="0" y="0" width="12" height="12"
style="stroke:black;fill:white" />

```

```

        </g>

```

```

    </g>

```

```

</svg>

```

10

#### skinComboBox\_Composite.svg

```

<?xml version="1.0"?>

```

```

<svg xmlns:dsvg="http://corel.org/dsvg">

```

```

15   <g id="skinComboBox">

```

```

        < u s e      i d = " T e x t B o x S k i n "
xlink:href="skinTextBox_ComboBox.svg#skinTextbox" x="0" y="0" />

```

```

        <use id="ButtonSkin" xlink:href="skinButton_ComboBox.svg#skinButton"
x="200" y="0" />

```

```

20   < u s e      i d = " D r o p D o w n S k i n "
xlink:href="skinListBox_Composite.svg#skinListBox" x="0" y="17" />

```

```

        <g id="label">

```

```

                <text id="skinComboBoxDefault_bodyText" x="0" y="-6"
fill="black"> </text>

```

```

25   <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="bottom"/>

```

```

        </g>
    </g>
</svg>

```

## 5 skinComboBox\_Default.svg

```

<?xml version="1.0"?>

<svg>

    <g id="skinComboBox">

        <g id="up" display="all">

10            <g id="skinComboBox_label">

                    <text x="0" y="-5" style="fill:#000000;font-
weight:normal;font-size:12;font-family:Tahoma"> </text>

                </g>

                <g id="skinComboBox_body" style="fill:none">

15                    <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

                    <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>

                    <path d="M 205 5 1 6 0 1 -3 6 z"

20 style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

                    <text id="skinComboBox_bodyText" x="5" y="13"
style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>

                </g>

                <g id="skinComboBox_bodyCover">

25                    <rect x="0" y="0" width="200" height="17" fill="none"
stroke="none" pointer-events="fill"/>

```

```

    </g>

    <g id="skinComboBox_dropDownButton">

        <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#000000;stroke-width:1"/>

5        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

        </g>

    </g>

10    <g id="down" display="none">

        <g id="skinComboBox_label">

            <text x="0" y="-5" style="fill:#000000;font-
weight:normal;font-size:12;font-family:Tahoma"> </text>

            </g>

15    <g id="skinComboBox_body" style="fill:none">

        <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

        <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>

20    <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

        <text id="skinComboBox_bodyText" x="5" y="13"
style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>

        </g>

25    <g id="skinComboBox_bodyCover">

```



```

        <rect x="0" y="0" width="200" height="17" fill="none"
stroke="none" pointer-events="fill"/>

    </g>

    <g id="skinComboBox_dropDownButton">
5        <rect x="200" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#199DBF;stroke-width:1"/>

        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

    </g>
10 </g>

    <g id="hover" display="none">

        <g id="skinComboBox_label">

            <text x="0" y="-5" style="fill:#000000;font-
15 weight:normal;font-size:12;font-family:Tahoma"> </text>

        </g>

        <g id="skinComboBox_body" style="fill:none">

            <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

20        <rect x="200" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3;stroke-width:1"/>

            <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>

            <text id="skinComboBox_bodyText" x="5" y="13"
25 style="fill:#676767;font-weight:normal;font-size:12;font-family:Tahoma"> </text>

        </g>

```

```

    <g id="skinComboBox_bodyCover">
        <rect fill="none" stroke="none" pointer-events="fill" x="0"
y="0" width="200" height="17"/>
    </g>
5    <g id="skinComboBox_dropDownButton">
        <rect x="200" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#199DBF;stroke-width:1"/>
        <path d="M 205 5 1 6 0 1 -3 6 z"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1"/>
10    </g>
    </g>

    <g id="skinComboBox_dropDownList">
        <g id="skinComboBox_dropDownList_top">
15            <line x1="0" y1="17" x2="0" y2="20"
style="stroke:#000000;stroke-width:1" />
            <line x1="217" y1="17" x2="217" y2="20"
style="stroke:#000000;stroke-width:1" />
            <rect x="1" y="18" width="215" height="2"
20 style="fill:#FFFFFF"/>
        </g>
        <g id="skinComboBox_dropDownList_middle">
            <line x1="0" y1="20" x2="0" y2="35"
style="stroke:#000000;stroke-width:1" />
25            <line x1="217" y1="20" x2="217" y2="35"
style="stroke:#000000;stroke-width:1" />

```

```

                    <rect x="1" y="20" width="215" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>
                </g>
                <g id="skinComboBox_dropDownList_bottom">
5                    <line x1="0" y1="35" x2="0" y2="38"
style="stroke:#000000;stroke-width:1" />
                    <line x1="217" y1="35" x2="217" y2="38"
style="stroke:#000000;stroke-width:1" />
10                <line x1="0" y1="38" x2="217" y2="38"
style="stroke:#000000;stroke-width:1" />
                    <rect x="1" y="36" width="215" height="2"
style="fill:#FFFFFF"/>
                </g>
15            </g>
        </g>
    </svg>

```

### skinComboBox\_Graphic.svg

```

20 <?xml version="1.0"?>
    <svg>
        <g id="skinComboBox">
            < u s e      i d = " T e x t B o x S k i n "
xlink:href="skinTextBox_ComboBox_Short.svg#skinTextbox" x="0" y="0" />
25        <use id="ButtonSkin" xlink:href="skinButton_ComboBox.svg#skinButton"
x="133" y="0" />

```

```

        < u s e      i d = " D r o p D o w n S k i n "
xlink:href="skinListBox_CompositeG.svg#skinListBox" x="0" y="17" />

```

```

    <g id="label">

```

```

        <text  id="skinComboBoxDefault_bodyText"  x="0"  y="-6"
5  fill="black"> </text>

```

```

        <dsvg:constraint  scaleImmunity="true"  hAlign="left"
vAlign="bottom"/>

```

```

    </g>

```

```

</g>

```

```

10 </svg>

```

### skinComboBox\_Graphic2.svg

```

<?xml version="1.0"?>

```

```

<svg>

```

```

15

```

```

    <g id="skinComboBox">

```

```

        < u s e      i d = " T e x t B o x S k i n "
xlink:href="skinTextBox_ComboBox_Short.svg#skinTextbox" x="0" y="0" />

```

```

        <use id="ButtonSkin" xlink:href="skinButton_ComboBox.svg#skinButton"
20 x="133" y="0" />

```

```

        < u s e      i d = " D r o p D o w n S k i n "
xlink:href="skinListBox_CompositeG2.svg#skinListBox" x="0" y="17" />

```

```

    <g id="label">

```

```

        <text  id="skinComboBoxDefault_bodyText"  x="0"  y="-6"
25 fill="black"> </text>

```

```

        <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="bottom"/>
    </g>
</g>
5
</svg>

```

### skinContextMenu\_Default.svg

```

<?xml version="1.0"?>
10 <svg xmlns:dsvg="http://corel.org/dsvg">
    <g id="skinContextMenu">
        <g id="skinContextMenu_top" display="all">
            <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-
width:1" />
15         <line x1="150" y1="17" x2="150" y2="20"
style="stroke:#000000;stroke-width:1" />
            <line x1="0" y1="17" x2="150" y2="17"
style="stroke:#000000;stroke-width:1" />
            <rect x="1" y="18" width="148" height="2" style="fill:#FFFFFF"/>
20         </g>
        <g id="skinContextMenu_middle" display="all">
            <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />
            <line x1="150" y1="20" x2="150" y2="35"
25 style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="20" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="32" id="label">

            <dsvg:constraint preserveAspectRatio="vertical" hAlign="left"
5   vAlign="bottom"/>

        </text>

    </g>

    <g id="skinContextMenu_middle_hover" display="none">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
10  width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1 isback="true"/>

15    <text x="3" y="32" id="label">

        <dsvg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>

        </text>

    </g>

20    <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

25    <rect x="1" y="20" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="32" id="label">

```

```

                    <dsvg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>
                </text>
            </g>
5          <g id="skinContextMenu_bottom" display="all">
                <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-
width:1" />
                <line x1="150" y1="35" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />
10          <line x1="0" y1="38" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />
                <rect x="1" y="36" width="215" height="2" style="fill:#FFFFFF"/>
            </g>
        </g>
15 </svg>

```

#### skinContextMenu\_Default2.svg

```

<?xml version="1.0"?>
<svg>
20 <g id="skinContextMenu">
        <g id="skinContextMenu_top" display="all">
                <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-
width:1" />
                <line x1="150" y1="17" x2="150" y2="20"
25 style="stroke:#000000;stroke-width:1" />

```

```

        <line x1="0" y1="17" x2="150" y2="17"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="18" width="148" height="2" style="fill:#FFFFFF"/>
    </g>
5    <g id="skinContextMenu_middle" display="all">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />
10    <rect x="1" y="20" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="32" id="label"> </text>

    </g>

    <g id="skinContextMenu_middle_hover" display="none">
15    <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
20    style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="32" id="label"> </text>

    </g>

    <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
25    width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

```



```

        <rect x="1" y="20" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```

```

        <text x="3" y="32" id="label"> </text>

```

```

    </g>

```

```

5      <g id="skinContextMenu_bottom" display="all">

```

```

        <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="150" y1="35" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />

```

```

10      <line x1="0" y1="38" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="36" width="215" height="2" style="fill:#FFFFFF"/>

```

```

    </g>

```

```

  </g>

```

```

15 </svg>

```

# skinContextMenu\_Graphic.svg

```

<?xml version="1.0"?>

```

```

<svg>

```

```

20   <g id="skinContextMenu">

```

```

        <g id="skinContextMenu_top" display="all">

```

```

        <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="150" y1="0" x2="150" y2="3"
25 style="stroke:#000000;stroke-width:1" />

```

```

        <line x1="0" y1="0" x2="150" y2="0" style="stroke:#000000;stroke-
width:1" />

```

```

        <rect x="1" y="1" width="148" height="2" style="fill:#FFFFFF"/>

```

```

    </g>

```

```

5    <g id="skinContextMenu_middle" display="all">

```

```

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

```

```

10    <rect x="1" y="3" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```

```

        <text x="3" y="21" id="label"> </text>

```

```

    </g>

```

```

    <g id="skinContextMenu_middle_hover" display="none">

```

```

15    <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="3" width="148" height="14"
20    style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```

```

        <text x="3" y="21" id="label"> </text>

```

```

    </g>

```

```

    <g id="skinContextMenu_middle_selected" display="none">

```

```

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
25    width:1" />

```

```

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="3" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```

```

        <text x="3" y="21" id="label"> </text>

```

```

    </g>

```

```

5      <g id="skinContextMenu_bottom" display="all">

```

```

        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="150" y1="18" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />

```

```

10      <line x1="0" y1="21" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="19" width="148" height="2" style="fill:#FFFFFF"/>

```

```

    </g>

```

```

15    </g>

```

```

    </svg>

```

### skinContextMenu\_Graphic2.svg

```

    <?xml version="1.0"?>

```

```

20    <svg>

```

```

        <g id="skinContextMenu">

```

```

            <g id="skinContextMenu_top" display="all">

```

```

                <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />

```

```

25      <line x1="133" y1="0" x2="133" y2="3"
style="stroke:#000000;stroke-width:1" />

```

```

<line x1="0" y1="0" x2="133" y2="0" style="stroke:#000000;stroke-
width:1" />

<rect x="1" y="1" width="131" height="2" style="fill:#FFFFFF"/>
</g>
5 <g id="skinContextMenu_middle" display="all">

    <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

    <line x1="133" y1="3" x2="133" y2="18"
style="stroke:#000000;stroke-width:1" />
10 <rect x="1" y="3" width="131" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>

    <text x="3" y="21" id="label"> </text>

    </g>

    <g id="skinContextMenu_middle_hover" display="none">
15 <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

    <line x1="133" y1="3" x2="133" y2="18"
style="stroke:#000000;stroke-width:1" />

    <rect x="1" y="3" width="131" height="14"
20 style="fill:#8888CC;stroke:#FFFFFF;stroke-width:1 isback="true"/>

    <text x="3" y="21" id="label"> </text>

    </g>

    <g id="skinContextMenu_middle_selected" display="none">

    <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
25 width:1" />

    <line x1="133" y1="3" x2="133" y2="18"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="3" width="131" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```

```

        <text x="3" y="21" id="label"> </text>

```

```

    </g>

```

```

5      <g id="skinContextMenu_bottom" display="all">

```

```

        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="133" y1="18" x2="133" y2="21"
style="stroke:#000000;stroke-width:1" />

```

```

10     <line x1="0" y1="21" x2="133" y2="21"
style="stroke:#000000;stroke-width:1" />

```

```

        <rect x="1" y="19" width="131" height="2" style="fill:#FFFFFF"/>

```

```

    </g>

```

```

15  </g>

```

```

</svg>

```

### skinContextMenu\_Graphic3.svg

```

20 <?xml version="1.0"?>

```

```

<svg>

```

```

    <g id="skinContextMenu">

```

```

        <g id="skinContextMenu_top" display="all">

```

```

        <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
25 width:1" />

```

```

        <line x1="233" y1="0" x2="233" y2="3"
style="stroke:#000000;stroke-width:1" />
        <line x1="0" y1="0" x2="233" y2="0" style="stroke:#000000;stroke-
width:1" />
5        <rect x="1" y="1" width="231" height="2" style="fill:#FFFFFF"/>
        </g>
        <g id="skinContextMenu_middle" display="all">
            <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />
10        <line x1="233" y1="3" x2="233" y2="18"
style="stroke:#000000;stroke-width:1" />
            <rect x="1" y="3" width="231" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>
            <text x="3" y="21" id="label"> </text>
15        </g>
        <g id="skinContextMenu_middle_hover" display="none">
            <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />
            <line x1="233" y1="3" x2="233" y2="18"
20 style="stroke:#000000;stroke-width:1" />
            <rect x="1" y="3" width="231" height="14"
style="fill:#8888CC;stroke:#FFFFFF;stroke-width:1 isback="true"/>
            <text x="3" y="21" id="label"> </text>
        </g>
25 <g id="skinContextMenu_middle_selected" display="none">
            <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

```

```

        <line x1="233" y1="3" x2="233" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="231" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1 isback="true"/>
5      <text x="3" y="21" id="label"> </text>

      </g>

      <g id="skinContextMenu_bottom" display="all">

        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />

10      <line x1="233" y1="18" x2="233" y2="21"
style="stroke:#000000;stroke-width:1" />

        <line x1="0" y1="21" x2="233" y2="21"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="19" width="231" height="2" style="fill:#FFFFFF"/>

15      </g>

      </g>

</svg>

20

skinContextMenu_ListBox.svg

<?xml version="1.0"?>

<svg xmlns:dsvg="http://corel.org/dsvg">

      <g id="skinContextMenu">

25      <g id="skinContextMenu_top" display="all">

```

```

        <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="0" x2="200" y2="3"
style="stroke:#000000;stroke-width:1" />
5      <line x1="0" y1="0" x2="200" y2="0" style="stroke:#000000;stroke-
width:1" />

        <rect x="1" y="1" width="198" height="2" style="fill:#FFFFFF"/>
    </g>
    <g id="skinContextMenu_middle" display="all">
10      <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="198" height="14"
15 style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="15" id="label">

            <dsvg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>

        </text>
20    </g>

    <g id="skinContextMenu_middle_hover" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
25 style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="198" height="14"
style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

```



```

        <text x="3" y="15" id="label">
            <svg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>
        </text>
5    </g>
    <g id="skinContextMenu_middle_selected" display="none">
        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />
        <line x1="200" y1="3" x2="200" y2="18"
10 style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="3" width="198" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1 isback="true"/>
        <text x="3" y="15" id="label">
            <svg:constraint preserveAspectRatio="vertical" hAlign="left"
15 vAlign="bottom"/>
        </text>
    </g>
    <g id="skinContextMenu_bottom" display="all">
        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
20 width:1" />
        <line x1="200" y1="18" x2="200" y2="21"
style="stroke:#000000;stroke-width:1" />
        <line x1="0" y1="21" x2="200" y2="21"
style="stroke:#000000;stroke-width:1" />
25 <rect x="1" y="19" width="198" height="2" style="fill:#FFFFFF"/>
    </g>

```

```

    </g>
</svg>

```

### skinContextMenu\_ListBox2.svg

```

5  <?xml version="1.0"?>
    <svg xmlns:dsvg="http://corel.org/dsvg">
        <g id="skinContextMenu">
            <g id="skinContextMenu_top" display="all">
                <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
10 width:1" />
                <line x1="200" y1="0" x2="200" y2="3"
                    style="stroke:#000000;stroke-width:1" />
                <line x1="0" y1="0" x2="200" y2="0" style="stroke:#000000;stroke-
width:1" />
15         <rect x="1" y="1" width="198" height="2" style="fill:#FFFFFF"/>
            </g>
            <g id="skinContextMenu_middle" display="all">
                <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />
20         <line x1="200" y1="3" x2="200" y2="18"
                    style="stroke:#000000;stroke-width:1" />
                <rect x="1" y="3" width="198" height="14"
                    style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>
                <text x="3" y="15" id="label">
25         <dsvg:constraint preserveAspectRatio="vertical" hAlign="left"
                    vAlign="bottom"/>

```

```

        </text>

    </g>

    <g id="skinContextMenu_middle_hover" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
5 width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="198" height="14"
style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1 isback="true"/>

10        <text x="3" y="15" id="label">

            <svg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>

        </text>

    </g>

15    <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />

20        <rect x="1" y="3" width="198" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="15" id="label">

            <svg:constraint preserveAspectRatio="vertical" hAlign="left"
vAlign="bottom"/>

25        </text>

    </g>

```

```

    <g id="skinContextMenu_bottom" display="all">
        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />
        <line x1="200" y1="18" x2="200" y2="21"
5 style="stroke:#000000;stroke-width:1" />
        <line x1="0" y1="21" x2="200" y2="21"
style="stroke:#000000;stroke-width:1" />
        <rect x="1" y="19" width="198" height="2" style="fill:#FFFFFF"/>
    </g>
10 </g>
</svg>

```

# skinContextMenu\_ListBoxG.svg

```

<?xml version="1.0"?>
15 <svg>
    <g id="skinContextMenu">
        <g id="skinContextMenu_top" display="all">
            <line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />
20 <line x1="150" y1="0" x2="150" y2="3"
style="stroke:#000000;stroke-width:1" />
            <line x1="0" y1="0" x2="150" y2="0" style="stroke:#000000;stroke-
width:1" />
            <rect x="1" y="1" width="148" height="2" style="fill:#FFFFFF"/>
25 </g>
        <g id="skinContextMenu_middle" display="all">

```

```

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />
5        <rect x="1" y="3" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_hover" display="none">
10        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="148" height="14"
15 style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
20 width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>
25        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_bottom" display="all">

```

```
<line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />
```

```
<line x1="150" y1="18" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />
```

```
5 <line x1="0" y1="21" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />
```

```
<rect x="1" y="19" width="215" height="2" style="fill:#FFFFFF"/>
```

```
</g>
```

```
</g>
```

```
10 </svg>
```

#### **skinContextMenu\_ListView.svg**

```
<?xml version="1.0"?>
```

```
<svg>
```

```
15 <g id="skinContextMenu">
```

```
<g id="skinContextMenu_top" display="all">
```

```
<line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />
```

```
20 <line x1="200" y1="0" x2="200" y2="3"
style="stroke:#000000;stroke-width:1" />
```

```
<line x1="0" y1="0" x2="200" y2="0" style="stroke:#000000;stroke-
width:1" />
```

```
<rect x="1" y="18" width="198" height="2" style="fill:#FFFFFF"/>
```

```
</g>
```

```
25 <g id="skinContextMenu_middle" display="all">
```

```

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />
5        <rect x="1" y="3" width="198" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="15" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_hover" display="none">
10        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="198" height="14"
15 style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="15" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
20 width:1" />

        <line x1="200" y1="3" x2="200" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="198" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>
25        <text x="3" y="15" id="label"> </text>

        </g>

        <g id="skinContextMenu_bottom" display="all">

```

```
<line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />
```

```
<line x1="200" y1="18" x2="200" y2="21"
style="stroke:#000000;stroke-width:1" />
```

```
5 <line x1="0" y1="21" x2="200" y2="21"
style="stroke:#000000;stroke-width:1" />
```

```
<rect x="1" y="19" width="198" height="2" style="fill:#FFFFFF"/>
```

```
</g>
```

```
</g>
```

```
10 </svg>
```

# skinContextMenu\_ListViewG.svg

```
<?xml version="1.0"?>
```

```
<svg>
```

```
15 <g id="skinContextMenu">
```

```
<g id="skinContextMenu_top" display="all">
```

```
<line x1="0" y1="0" x2="0" y2="3" style="stroke:#000000;stroke-
width:1" />
```

```
20 <line x1="150" y1="0" x2="150" y2="3"
style="stroke:#000000;stroke-width:1" />
```

```
<line x1="0" y1="0" x2="150" y2="0" style="stroke:#000000;stroke-
width:1" />
```

```
<rect x="1" y="1" width="148" height="2" style="fill:#FFFFFF"/>
```

```
</g>
```

```
25 <g id="skinContextMenu_middle" display="all">
```



```

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />
5        <rect x="1" y="3" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_hover" display="none">
10        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="148" height="14"
15 style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1 isback="true"/>

        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="3" x2="0" y2="18" style="stroke:#000000;stroke-
20 width:1" />

        <line x1="150" y1="3" x2="150" y2="18"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="3" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1 isback="true"/>
25        <text x="3" y="5" id="label"> </text>

        </g>

        <g id="skinContextMenu_bottom" display="all">

```

```

        <line x1="0" y1="18" x2="0" y2="21" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="18" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />
5        <line x1="0" y1="21" x2="150" y2="21"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="19" width="215" height="2" style="fill:#FFFFFF"/>

        </g>

    </g>
10 </svg>

```

#### skinContextMenu\_vp.svg

```

<?xml version="1.0"?>

<svg>
15 <g id="skinContextMenu">

    <g id="skinContextMenu_top" display="all">

        <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="17" x2="150" y2="20"
20 style="stroke:#000000;stroke-width:1" />

        <line x1="0" y1="17" x2="150" y2="17"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="18" width="148" height="2" style="fill:#FFFFFF"/>

        </g>

25 <g id="skinContextMenu_middle" display="all">

```

```

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />
5        <rect x="1" y="20" width="148" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="32" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_hover" display="none">
10        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
15 style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

        <text x="3" y="32" id="label"> </text>

        </g>

        <g id="skinContextMenu_middle_selected" display="none">

        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
20 width:1" />

        <line x1="150" y1="20" x2="150" y2="35"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="20" width="148" height="14"
style="fill:#FF0000;stroke:#FFFFFF;stroke-width:1" isback="true"/>

25        <text x="3" y="32" id="label"> </text>

        </g>

        <g id="skinContextMenu_bottom" display="all">

```

```

        <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-
width:1" />

        <line x1="150" y1="35" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />
5        <line x1="0" y1="38" x2="150" y2="38"
style="stroke:#000000;stroke-width:1" />

        <rect x="1" y="36" width="215" height="2" style="fill:#FFFFFF"/>

        </g>

    </g>
10 </svg>

```

#### skinListbox\_Composite.svg

```

<?xml version="1.0"?>

<svg>
15 <g id="skinListBox">

        < u s e      i d = " M e n u S k i n "
xlink:href="skinContextMenu_ListBox.svg#skinContextMenu" x="0" y="0" />

        < u s e      i d = " S c r o l l S k i n "
xlink:href="skinScrollBar_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>

20 <g id="label">

        <text id="skinListBox_label" x="0" y="-6" fill="black"> </text>

        </g>

    </g>

</svg>

```

25

#### skinListbox\_CompositeG.svg

```

<?xml version="1.0"?>

<svg>

    <g id="skinListBox">

        < u s e      i d = " M e n u S k i n "
5  xlink:href="skinContextMenu_Graphic.svg#skinContextMenu" x="0" y="0" />

        < u s e      i d = " S c r o l l S k i n "
xlink:href="skinScrollBar_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>

        <g id="label">

            <text id="skinListBox_label" x="0" y="-6" fill="black"> </text>

10        </g>

    </g>

</svg>

```

# skinListbox\_CompositeG2.svg

```

15 <?xml version="1.0"?>

<svg>

    <g id="skinListBox">

        < u s e      i d = " M e n u S k i n "
20 xlink:href="skinContextMenu_Graphic2.svg#skinContextMenu" x="0" y="0" />

        < u s e      i d = " S c r o l l S k i n "
xlink:href="skinScrollBar_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>

        <g id="label">

            <text id="skinListBox_label" x="0" y="-6" fill="black"> </text>

25        </g>

    </g>

</svg>

```

</svg>

### skinListbox\_CompositeG3.svg

5 <?xml version="1.0"?>

<svg>

<g id="skinListBox">

< u s e i d = " M e n u S k i n "

10 xlink:href="skinContextMenu\_Graphic3.svg#skinContextMenu" x="0" y="0" />

< u s e i d = " S c r o l l S k i n "

xlink:href="skinScrollBar\_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>

<g id="label">

<text id="skinListBox\_label" x="0" y="-6" fill="black"> </text>

15 </g>

</g>

</svg>

### 20 skinListbox\_Default.svg

<?xml version="1.0"?>

<svg xmlns:dsvg="http://corel.org/dsvg" >

<g id="skinButton">

<g id="label">

```

        <text x="50" y="13" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

    </g>

    <g id="up" display="all" style="stroke-width:1">
5        <rect x="0" y="0" width="17" height="17"
style="fill:#000000;stroke:#A3A3A3"/>

        <path d="M551601-36z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
        <svg:constraint preserveAspectRatio="vertical" hAlign="right"/>
    </g>

10    <g id="down" display="none">

        <rect x="0" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#A3A3A3"/>

        <path d="M551601-36z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
        <svg:constraint preserveAspectRatio="vertical" hAlign="right"/>

15    </g>

    <g id="hover" display="none">

        <rect x="0" y="0" width="17" height="17"
style="fill:#199DBF;stroke:#A3A3A3"/>

        <path d="M551601-36z" style="fill:#FFFFFF;stroke:#FFFFFF"/>
20    <svg:constraint preserveAspectRatio="vertical" hAlign="right"/>

    </g>

</g>

</svg>

25 skinListbox_Short.svg

<?xml version="1.0"?>

```

```

<svg>
  <g id="skinListBox">
    <use id="Text Box Skin"
      xlink:href="skinTextBox_ListBox_Short.svg#skinTextbox" x="0" y="0" />
5    <use id="ButtonSkin" xlink:href="skinButton_ListBox.svg#skinButton"
      x="100" y="0" />
    <use id="Drop Down Skin"
      xlink:href="skinListBox_Short.svg#skinDropDown" x="0" y="0" />
    <g id="label">
10    <text id="skinListBoxDefault_bodyText" x="0" y="-6" fill="black">
      </text>
    </g>
  </g>
15 <g id="skinDropDown">
  <g id="skinDropDown_top">
    <line x1="0" y1="17" x2="0" y2="20" style="stroke:#000000;stroke-
width:1" />
    <line x1="117" y1="17" x2="117" y2="20"
20 style="stroke:#000000;stroke-width:1" />
    <rect x="1" y="18" width="115" height="2" style="fill:#FFFFFF"/>
  </g>
  <g id="skinDropDown_middle">
    <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
25 width:1" />

```



```

                    <line x1="117" y1="20" x2="117" y2="35"
style="stroke:#000000;stroke-width:1" />

                    <rect x="1" y="20" width="115" height="14"
style="fill:#FFFFFF;stroke:#FFFFFF;stroke-width:1" isback="true"/>
5                    <text x="3" y="32" id="label"> </text>

                    </g>

                    <g id="skinDropDown_middle_hover" display="none">

                        <line x1="0" y1="20" x2="0" y2="35" style="stroke:#000000;stroke-
width:1" />

10                    <line x1="117" y1="20" x2="117" y2="35"
style="stroke:#000000;stroke-width:1" />

                    <rect x="1" y="20" width="115" height="14"
style="fill:#CCCCCC;stroke:#FFFFFF;stroke-width:1" isback="true"/>

                    <text x="3" y="32" id="label"> </text>

15                    </g>

                    <g id="skinDropDown_bottom">

                        <line x1="0" y1="35" x2="0" y2="38" style="stroke:#000000;stroke-
width:1" />

                        <line x1="117" y1="35" x2="117" y2="38"
20 style="stroke:#000000;stroke-width:1" />

                        <line x1="0" y1="38" x2="117" y2="38"
style="stroke:#000000;stroke-width:1" />

                        <rect x="1" y="36" width="115" height="2" style="fill:#FFFFFF"/>

                    </g>

25 </g>

</svg>

```

**skinListView\_Composite.svg**

```
<?xml version="1.0"?>
```

```
<svg>
```

```
  <g id="skinListView">
```

```
5      < u s e      i d = " M e n u S k i n "
xlink:href="skinContextMenu_ListBox.svg#skinContextMenu" x="0" y="0" />
```

```
      < u s e      i d = " S c r o l l S k i n "
xlink:href="skinScrollBar_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>
```

```
      <g id="label">
```

```
10      <text id="skinListBox_label" x="0" y="-6" fill="black"> </text>
```

```
      </g>
```

```
    </g>
```

```
  </svg>
```

**15 skinListView\_CompositeG.svg**

```
<?xml version="1.0"?>
```

```
<svg>
```

```
  <g id="skinListView">
```

```
      < u s e      i d = " M e n u S k i n "
20  xlink:href="skinContextMenu_Graphic.svg#skinContextMenu" x="0" y="0" />
```

```
      < u s e      i d = " S c r o l l S k i n "
xlink:href="skinScrollBar_ListBox.svg#skinScrollBarSmall" x="0" y="0"/>
```

```
      <g id="label">
```

```
      <text id="skinListBox_label" x="0" y="-6" fill="black"> </text>
```

```
25      </g>
```

```
    </g>
```

```
</svg>
```

### skinRadioButton\_Default.svg

```
<?xml version="1.0"?>
```

```
5 <svg>
```

```
    <g id="skinRadioButton">
```

```
        <g id="label">
```

```
            <text x="16" y="5" style="font-size:12;fill:black"> </text>
```

```
        </g>
```

```
10    <g id="up" display="all">
```

```
        <circle cx="5" y="5" r="7" style="stroke:black;fill:white" />
```

```
    </g>
```

```
    <g id="down" display="none">
```

```
        <circle cx="5" y="5" r="7" style="stroke:black;fill:white" />
```

```
15    <circle cx="5" y="5" r="4" style="stroke:black;fill:black" />
```

```
    </g>
```

```
    <g id="hover" display="none">
```

```
        <circle cx="5" y="5" r="7" style="stroke:black;fill:white" />
```

```
    </g>
```

```
20    <g id="disabled" style="opacity:0.5" display="none">
```

```
        <circle cx="5" y="5" r="7" style="stroke:black;fill:white" />
```

```
    </g>
```

```
    </g>
```

```
</svg>
```

```
25
```

**skinScrollBar\_Default.svg**

```
<?xml version="1.0"?>
```

```
<svg>
```

```
  <g id="skinScrollBar">
```

```
5      <use id="SliderSkin" xlink:href="skinScrollBar_Default.svg#SliderSkin"
x="0" y="0"/>
```

```
      <use id="ButtonLeftSkin"
xlink:href="skinScrollBar_Default.svg#ButtonLeftSkin" x="0" y="0"/>
```

```
      <use id="ButtonRightSkin"
10  xlink:href="skinScrollBar_Default.svg#ButtonRightSkin" x="0" y="0"/>
```

```
      <g id="corner">
```

```
        <rect fill="rgb(212 208 200)" x="0" y="0" width="16" height="16"
```

```
      />
```

```
    </g>
```

```
15  </g>
```

```
    <g id="ButtonLeftSkin">
```

```
      <g id="up" fill="none" stroke-linecap="square" shape-
rendering="optimizeSpeed" display="all">
```

```
        <rect fill="rgb(212 208 200)" width="16" height="16" />
```

```
20    <line x1="15" y1="0" x2="15" y2="15" style="stroke:#404040"/>
```

```
    <line x1="0" y1="15" x2="15" y2="15" style="stroke:#404040"/>
```

```
    <line x1="1" y1="1" x2="13" y2="1" style="stroke:#FFFFFF"/>
```

```
    <line x1="1" y1="1" x2="1" y2="13" style="stroke:#FFFFFF"/>
```

```
    <line x1="1" y1="14" x2="14" y2="14" style="stroke:#808080"/>
```

```
25    <line x1="14" y1="1" x2="14" y2="14" style="stroke:#808080"/>
```

```
    <path fill="black" d="M9.5 5.5l-3 3l3 3z" />
```

```

    </g>

    <g id="down" fill="none" stroke-linecap="square" shape-
rendering="optimizeSpeed" display="none">

        <rect fill="rgb(212 208 200)" width="16" height="16" />
5      <rect fill="none" style="stroke:#808080" width="16" height="16"/>

        <path fill="black" d="M10.5 6.5l-3 3l3 3z" />

    </g>

</g>

<g id="SliderSkin">
10   <g id="up" display="all">

        <g id="body">

            <rect id="SliderSkinBar" fill="#E6E6E6" x="0" y="0"
width="400" height="16">

                </rect>

15         </g>

        <g id="SliderSkinThumb">

            <rect id="sliderThumbBar" style="fill:rgb(212 208
200);stroke:#000000;stroke-width:0" x="0" y="0" width="400" height="16"/>

                <line x1="0" x2="400" y1="15" y2="15"
20 style="stroke:#404040"/>

                <line x1="1" x2="398" y1="1" y2="1"
style="stroke:#FFFFFF"/>

            </g>

            <pattern id="barfill" width="2" height="2"
25 patternUnits="userSpaceOnUse" shape-rendering="optimizeSpeed">

                <rect x="0" y="0" width="2" height="2" fill="white"/>

```

```

    <rect x="0" y="0" width="1" height="1" fill="rgb(212 208
200)"/>

    <rect x="1" y="1" width="1" height="1" fill="rgb(212 208
200)"/>

5      </pattern>

      </g>

      <g id="mask" >

        <rect id="sliderMask" x="0" y="0" width="400" height="16"
10    opacity="0"/>

      </g>

    </g>

    <g id="ButtonRightSkin">

      <g id="up" fill="none" stroke-linecap="square" shape-
15    rendering="optimizeSpeed" display="all">

        <rect fill="rgb(212 208 200)" width="16" height="16" />

        <line x1="15" y1="0" x2="15" y2="15" style="stroke:#404040"/>

        <line x1="0" y1="15" x2="15" y2="15" style="stroke:#404040"/>

        <line x1="1" y1="1" x2="13" y2="1" style="stroke:#FFFFFF"/>

20    <line x1="1" y1="1" x2="1" y2="13" style="stroke:#FFFFFF"/>

        <line x1="1" y1="14" x2="14" y2="14" style="stroke:#808080"/>

        <line x1="14" y1="1" x2="14" y2="14" style="stroke:#808080"/>

        <path fill="black" d="M6.5 5.5l3 3l-3 3z" />

      </g>

25    <g id="down" fill="none" stroke-linecap="square" shape-
    rendering="optimizeSpeed" display="none">

```

```
<rect fill="rgb(212 208 200)" width="16" height="16" />
```

```
<rect fill="none" style="stroke:#808080" width="16" height="16"/>
```

```
<path fill="black" d="M7.5 6.5l3 3l-3 3z" />
```

```
</g>
```

```
5 </g>
```

```
<g id="skinScrollBarSmall">
```

```
10 < u s e      i d = " S l i d e r S k i n "
xlink:href="skinScrollBar_Default.svg#SliderSkinSmall" x="0" y="0"/>
```

```
< u s e      i d = " B u t t o n L e f t S k i n "
xlink:href="skinScrollBar_Default.svg#ButtonLeftSkinSmall" x="0" y="0"/>
```

```
< u s e      i d = " B u t t o n R i g h t S k i n "
15 xlink:href="skinScrollBar_Default.svg#ButtonRightSkinSmall" x="0" y="0"/>
```

```
<g id="corner">
```

```
<rect x="0" y="0" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>
```

```
</g>
```

```
20 </g>
```

```
<g id="ButtonLeftSkinSmall">
```

```
<g id="up" display="all">
```

```
<rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
25 miterlimit:2.6131;"/>
```

```

        <rect x="0" y="0" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

        <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

5          <path d="M 8 2 1 -6 3 1 6 3 z"
style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

        </g>

        <g id="down" display="none">

            <rect x="1" y="1" width="10" height="10" style="fill-
10 rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

            <rect x="1" y="1" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

            <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
15 size:12;text-anchor:middle"> </text>

            <path d="M 9 3 1 -6 3 1 6 3 z"
style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

        </g>

    </g>

20    <g id="SliderSkinSmall">

        <g id="up" display="all">

            <rect id="SkinSliderSmallBorder" x="0" y="0" width="200"
height="10" opacity="0"/>

            <g id="body">

25                <rect id="SkinSliderSmallBase" style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="1" y="1" width="444" height="10"/>

```



```

        <rect id="SliderSkinSmallBar"
style="fill:#FFFFFF;stroke:#000000" x="0" y="0" width="444" height="10"/>

        </g>

        <g id="SliderSkinSmallThumb">
5          <rect id="sliderThumbBar"
style="fill:#B3B3B3;stroke:#000000;stroke-width:0" x="0" y="0" width="10" height="10"/>

          </g>

          <g id="tickMajor" display="none">

              <text x="0" y="11" id="tickLabelMajor"
10  xml:space="preserve" style="font-face:Arial;font-size:8;fill:black;stroke:1;text-
anchor:middle"></text>

              </g>

          </g>

          <g id="mask" >

15          <rect id="sliderMask" x="" y="0" width="444" height="18"
opacity="0"/>

          </g>

        </g>

        <g id="ButtonRightSkinSmall">

20          <g id="up" display="all">

              <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

              <rect x="0" y="0" width="10" height="10"
25  style="fill:#FFFFFF;stroke:#000000"/>

              <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

```

```

        <path d="M 2 2 1 6 3 1 -6 3 z"
style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

```

```

    </g>

```

```

    <g id="down" display="none">

```

```

5      <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

```

```

        <rect x="1" y="1" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

```

```

10      <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

```

```

        <path d="M 3 3 1 6 3 1 -6 3 z"
style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

```

```

    </g>

```

```

15  </g></svg>

```

### skinScrollBar\_ListBox.svg

```

<?xml version="1.0"?>

```

```

<svg>

```

```

20  <g id="skinScrollBarSmall">

```

```

        < u s e      i d = " S l i d e r S k i n "
xlink:href="skinScrollBar_Default.svg#SliderSkinSmall" x="0" y="0"/>

```

```

        < u s e      i d = " B u t t o n L e f t S k i n "
xlink:href="skinScrollBar_Default.svg#ButtonLeftSkinSmall" x="0" y="0"/>

```

```

25  < u s e      i d = " B u t t o n R i g h t S k i n "
xlink:href="skinScrollBar_Default.svg#ButtonRightSkinSmall" x="0" y="0"/>

```

```

        <g id="corner">

```

```

        <rect x="0" y="0" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000" display="none"/>

    </g>

</g>

5   <g id="ButtonLeftSkinSmall">

        <g id="up" display="all">

                <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

10      <rect x="0" y="0" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

                <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

                <path d="M 8 2 1 -6 3 1 6 3 z"
15 style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

        </g>

        <g id="down" display="none">

                <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
20 miterlimit:2.6131;"/>

                <rect x="1" y="1" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

                <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

25      <path d="M 9 3 1 -6 3 1 6 3 z"
style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

        </g>

```

```

    </g>

    <g id="SliderSkinSmall">

        <g id="up" display="all">

            <rect id="SkinSliderSmallBorder" x="0" y="0" width="200"
5 height="10" opacity="0"/>

            <g id="body">

                <rect id="SkinSliderSmallBase" style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="1" y="1" width="444" height="10"/>

10         <rect id="SliderSkinSmallBar"
style="fill:#FFFFFF;stroke:#000000" x="0" y="0" width="444" height="10"/>

                </g>

                <g id="SliderSkinSmallThumb">

                    <rect id="sliderThumbBar"
15 style="fill:#B3B3B3;stroke:#000000;stroke-width:0" x="0" y="0" width="10" height="10"/>

                    </g>

                    <g id="tickMajor" display="none">

                        <text x="0" y="11" id="tickLabelMajor"
xml:space="preserve" style="font-face:Arial;font-size:8;fill:black;stroke:1;text-
20 anchor:middle"></text>

                    </g>

                </g>

                <g id="mask">

                    <rect id="sliderMask" x="" y="0" width="444" height="18"
25 opacity="0"/>

                </g>

```

</g>

<g id="ButtonRightSkinSmall">

<g id="up" display="all">

5 <rect x="1" y="1" width="10" height="10" style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"/>

<rect x="0" y="0" width="10" height="10" style="fill:#FFFFFF;stroke:#000000"/>

10 <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

<path d="M 2 2 1 6 3 1 -6 3 z" style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

</g>

<g id="down" display="none">

15 <rect x="1" y="1" width="10" height="10" style="fill-rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"/>

<rect x="1" y="1" width="10" height="10" style="fill:#FFFFFF;stroke:#000000"/>

20 <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-size:12;text-anchor:middle"> </text>

<path d="M 3 3 1 6 3 1 -6 3 z" style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

</g>

25 </g></svg>

skinScrollBar\_ListView.svg

```

<?xml version="1.0"?>

<svg>

  <g id="skinScrollBarSmall">

    < u s e      i d = " S l i d e r S k i n "
5    xlink:href="skinScrollBar_Default.svg#SliderSkinSmall" x="0" y="0"/>

    < u s e      i d = " B u t t o n L e f t S k i n "
    xlink:href="skinScrollBar_Default.svg#ButtonLeftSkinSmall" x="0" y="0"/>

    < u s e      i d = " B u t t o n R i g h t S k i n "
    xlink:href="skinScrollBar_Default.svg#ButtonRightSkinSmall" x="0" y="0"/>

10    <g id="corner">

      <rect x="0" y="0" width="10" height="10"
      style="fill:#FFFFFF;stroke:#000000" display="none"/>

      </g>

    </g>

15    <g id="ButtonLeftSkinSmall">

      <g id="up" display="all">

        <rect x="1" y="1" width="10" height="10" style="fill-
        rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
        miterlimit:2.6131;"/>

20        <rect x="0" y="0" width="10" height="10"
        style="fill:#FFFFFF;stroke:#000000"/>

        <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
        size:12;text-anchor:middle"> </text>

        <path d="M 8 2 1 -6 3 1 6 3 z"

25 style="fill:#000000;stroke:#FF0000;stroke-width:0"/>

      </g>

      <g id="down" display="none">

```

```

        <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

        <rect x="1" y="1" width="10" height="10"
5 style="fill:#FFFFFF;stroke:#000000"/>

        <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

        <path d="M 9 3 1 -6 3 1 6 3 z"
style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

10 </g>

    </g>

    <g id="SliderSkinSmall">

        <g id="up" display="all">

            <rect id="SkinSliderSmallBorder" x="0" y="0" width="200"
15 height="10" opacity="0"/>

            <g id="body">

                <rect id="SkinSliderSmallBase" style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="1" y="1" width="444" height="10"/>

20 <rect id="SliderSkinSmallBar"
style="fill:#FFFFFF;stroke:#000000" x="0" y="0" width="444" height="10"/>

            </g>

            <g id="SliderSkinSmallThumb">

                <rect id="sliderThumbBar"
25 style="fill:#B3B3B3;stroke:#000000;stroke-width:0" x="0" y="0" width="10" height="10"/>

            </g>

            <g id="tickMajor" display="none">

```

```

    <text x="0" y="11" id="tickLabelMajor"
xml:space="preserve" style="font-face:Arial;font-size:8;fill:black;stroke:1;text-
anchor:middle"></text>

    </g>
5    </g>
    <g id="mask" >
        <rect id="sliderMask" x="" y="0" width="444" height="18"
opacity="0"/>
    </g>
10 </g>
    <g id="ButtonRightSkinSmall">
        <g id="up" display="all">
            <rect x="1" y="1" width="10" height="10" style="fill-
rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
15 miterlimit:2.6131;"/>
            <rect x="0" y="0" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>
            <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>
20 <path d="M 2 2 1 6 3 1 -6 3 z"
style="fill:#000000;stroke:#FF0000;stroke-width:0"/>
        </g>
        <g id="down" display="none">
            <rect x="1" y="1" width="10" height="10" style="fill-
25 rule:evenodd;clip-rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-
miterlimit:2.6131;"/>

```



```

        <rect x="1" y="1" width="10" height="10"
style="fill:#FFFFFF;stroke:#000000"/>

```

```

        <text x="5" y="5" style="font-family:'Tahoma';stroke:none;font-
size:12;text-anchor:middle"> </text>

```

```

5         <path d="M 3 3 1 6 3 1 -6 3 z"
style="fill:#FF0000;stroke:#FF0000;stroke-width:0"/>

```

```

        </g>

```

```

    </g></svg>

```

# 10 skinSlider\_Default.svg

```

<?xml version="1.0"?>

```

```

<svg xmlns:dsvg="http://corel.org/dsvg">

```

```

    <g id="skinSlider">

```

```

        <g id="label">

```

```

15        <text id="skinComboBoxDefault_bodyText" x="0" y="-6"
fill="black"> </text>

```

```

        <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="bottom"/>

```

```

        </g>

```

```

20    <g id="up" display="all">

```

```

        <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
fill="none" stroke="none" pointer-events="fill"/>

```

```

        <g id="body">

```

```

        <rect id="skinSliderBase" fill="#EEEEEE" stroke="none"
25 x="0" y="7" width="200" height="4"/>

```

```

        <rect id="skinSliderBar" fill="#EEEEEE" stroke="none"
x="0" y="7" width="200" height="4"/>
        <line x1="0" y1="7" x2="200" y2="7"
style="stroke:#B3B3B3"/>
5      </g>
      <g id="skinSliderThumb">
        <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
        <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>
        <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
10 width="6" height="12"/>
        <line x1="0" y1="3" x2="0" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
        <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>
15      <line x1="6" y1="3" x2="6" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
      </g>
      <g id="tickMajor" display="none">
        <line x1="0" y1="7" x2="0" y2="17"
20 style="stroke:#B3B3B3"/>
        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke-width:1"> </text>
      </g>
      <g id="tickMinor" display="none">
25      <line x1="0" y1="7" x2="0" y2="11"
style="stroke:#B3B3B3"/>
      </g>

```

</g>

<g id="down" display="none">

<rect id="skinSliderBorder" x="0" y="0" width="200" height="18"

5 fill="none" stroke="none" pointer-events="fill"/>

<g id="body">

<rect id="skinSliderBase" fill="#EEEEEE" stroke="none"  
x="0" y="7" width="200" height="4"/>

<rect id="skinSliderBar" fill="#EEEEEE" stroke="none"  
10 x="0" y="7" width="200" height="4"/>

<line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>

</g>

<g id="skinSliderThumb">

<circle cx="3" cy="3" r="3"  
15 style="fill:#199DBF;stroke:none"/>

<circle cx="3" cy="15" r="3"  
style="fill:#199DBF;stroke:none"/>

<rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"  
width="6" height="12"/>

<line x1="0" y1="3" x2="0" y2="15"  
20 style="stroke:#B3B3B3;stroke-width:1"/>

<line x1="1.5" y1="3" x2="1.5" y2="15"  
style="stroke:white;stroke-width:2"/>

<line x1="6" y1="3" x2="6" y2="15"  
25 style="stroke:#B3B3B3;stroke-width:1"/>

</g>

<g id="tickMajor" display="none">

```

        <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>

        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:black;stroke-width:1"> </text>

    </g>

5    <g id="tickMinor" display="none">

        <line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>

    </g>

</g>

10    <g id="hover" display="none">

        <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
fill="none" stroke="none" pointer-events="fill"/>

        <g id="body">

            <rect id="skinSliderBase" fill="#EEEEEE" stroke="none"
15    x="0" y="7" width="200" height="4"/>

            <rect id="skinSliderBar" fill="#EEEEEE" stroke="none"
x="0" y="7" width="200" height="4"/>

            <line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>

        </g>

20    <g id="skinSliderThumb">

        <circle cx="3" cy="3" r="3"
style="fill:#199DBF;stroke:none"/>

        <circle cx="3" cy="15" r="3"
style="fill:#199DBF;stroke:none"/>

25    <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>

```

```

                    <line x1="0" y1="3" x2="0" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>

                    <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>
5                    <line x1="6" y1="3" x2="6" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>

                </g>

                <g id="tickMajor" display="none">

                    <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>
10                    <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:black;stroke-width:1"> </text>

                </g>

                <g id="tickMinor" display="none">

                    <line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>
15                </g>

            </g>

            <g id="disabled" display="none">

                <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
20 fill="none" stroke="none" pointer-events="fill"/>

                <g id="body">

                    <rect id="skinSliderBase" fill="#EEEEEE" stroke="none"
x="0" y="7" width="200" height="4"/>

                    <rect id="skinSliderBar" fill="#EEEEEE" stroke="none"
25 x="0" y="7" width="200" height="4"/>

                    <line x1="0" y1="7" x2="200" y2="7" stroke="#B3B3B3"/>

```

```

    </g>

    <g id="skinSliderThumb">

        <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
        <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>
5        <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>

        <line x1="0" y1="3" x2="0" y2="15"
style="stroke:B3B3B3;stroke-width:1"/>

        <line x1="1.5" y1="3" x2="1.5" y2="15"
10 style="stroke:white;stroke-width:2"/>

        <line x1="6" y1="3" x2="6" y2="15" style="stroke:black" />

    </g>

    <g id="tickMajor" display="none">

        <line x1="0" y1="7" x2="0" y2="17"
15 style="stroke:#B3B3B3"/>

        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke-width:1"> </text>

    </g>

    <g id="tickMinor" display="none">

20        <line x1="0" y1="7" x2="0" y2="11" stroke="#B3B3B3"/>

    </g>

</g>

    <g id="mask" >

25        <rect id="skinSliderMask" x="-4" y="0" width="208" height="18"
fill="none" stroke="none" pointer-events="fill"/>

```

</g>

</g>

</svg>

## 5 skinSlider\_Windows.svg

<?xml version="1.0"?>

<svg xmlns:dsvg="http://corel.org/dsvg">

<g id="skinSlider">

<g id="label">

10 <text id="skinComboBoxDefault\_bodyText" x="0" y="-6"  
fill="black"> </text>

<dsvg:constraint scaleImmunity="true" hAlign="left"  
vAlign="bottom"/>

</g>

15 <g id="up" display="all">

<rect id="skinSliderBorder" x="0" y="0" width="200" height="18"  
fill="none" stroke="none" pointer-events="fill"/>

<g id="body">

20 <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none"  
x="0" y="7" width="200" height="4"/>

<rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none"  
x="0" y="7" width="200" height="4"/>

<line x1="0" y1="7" x2="200" y2="7"  
style="stroke:#B3B3B3"/>

25 </g>

<g id="skinSliderThumb">

```

        <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
        <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>
        <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>
5         <line x1="0" y1="3" x2="0" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
        <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>
        <line x1="6" y1="3" x2="6" y2="15"
10 style="stroke:#B3B3B3;stroke-width:1"/>
        </g>
        <g id="tickMajor" display="none">
        <line x1="0" y1="7" x2="0" y2="17"
style="stroke:#B3B3B3"/>
15 <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke:1;text-anchor:middle"></text>
        </g>
        <g id="tickMinor" display="none">
        <line x1="0" y1="7" x2="0" y2="11"
20 style="stroke:#B3B3B3"/>
        </g>
        </g>

        <g id="down" display="none">
25 <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
fill="none" stroke="none" pointer-events="fill"/>

```



```

    <g id="body">
        <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none"
x="0" y="7" width="200" height="4"/>
        <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none"
5 x="0" y="7" width="200" height="4"/>
        <line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>
    </g>
    <g id="skinSliderThumb">
        <circle cx="3" cy="3" r="3"
10 style="fill:#199DBF;stroke:none"/>
        <circle cx="3" cy="15" r="3"
style="fill:#199DBF;stroke:none"/>
        <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>
15 <line x1="0" y1="3" x2="0" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
        <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>
        <line x1="6" y1="3" x2="6" y2="15"
20 style="stroke:#B3B3B3;stroke-width:1"/>
    </g>
    <g id="tickMajor" display="none">
        <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>
        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
25 style="font-face:Arial;font-size:8;fill:black;stroke:1;text-anchor:middle"></text>
    </g>
    <g id="tickMinor" display="none">

```

```
<line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>
```

```
</g>
```

```
</g>
```

```
5      <g id="hover" display="none">
```

```
      <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
fill="none" stroke="none" pointer-events="fill"/>
```

```
      <g id="body">
```

```
10      <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none"
x="0" y="7" width="200" height="4"/>
```

```
      <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none"
x="0" y="7" width="200" height="4"/>
```

```
      <line x1="0" y1="7" x2="200" y2="7" style="stroke:black"/>
```

```
      </g>
```

```
15      <g id="skinSliderThumb">
```

```
      <circle cx="3" cy="3" r="3"
style="fill:#199DBF;stroke:none"/>
```

```
      <circle cx="3" cy="15" r="3"
style="fill:#199DBF;stroke:none"/>
```

```
20      <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>
```

```
      <line x1="0" y1="3" x2="0" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
```

```
25      <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>
```

```
      <line x1="6" y1="3" x2="6" y2="15"
style="stroke:#B3B3B3;stroke-width:1"/>
```

```

    </g>

    <g id="tickMajor" display="none">
        <line x1="0" y1="7" x2="0" y2="17" style="stroke:black"/>
        <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
5    style="font-face:Arial;font-size:8;fill:black;stroke:1;text-anchor:middle"></text>
    </g>

    <g id="tickMinor" display="none">
        <line x1="0" y1="7" x2="0" y2="11" style="stroke:black"/>
    </g>
10 </g>

    <g id="disabled" display="none">
        <rect id="skinSliderBorder" x="0" y="0" width="200" height="18"
        fill="none" stroke="none" pointer-events="fill"/>
15 <g id="body">
        <rect id="skinSliderBase" style="fill:#FFFFFF;stroke:none"
        x="0" y="7" width="200" height="4"/>
        <rect id="skinSliderBar" style="fill:#FFFFFF;stroke:none"
        x="0" y="7" width="200" height="4"/>
20 <line x1="0" y1="7" x2="200" y2="7"
        style="stroke:#B3B3B3"/>
    </g>

    <g id="skinSliderThumb">
        <circle cx="3" cy="3" r="3" style="fill:black;stroke:none"/>
25 <circle cx="3" cy="15" r="3" style="fill:black;stroke:none"/>

```

```

        <rect style="fill:rgb(230,230,230);stroke:none" x="0" y="3"
width="6" height="12"/>

        <line x1="0" y1="3" x2="0" y2="15"
style="#stroke:B3B3B3;stroke-width:1"/>
5        <line x1="1.5" y1="3" x2="1.5" y2="15"
style="stroke:white;stroke-width:2"/>

        <line x1="6" y1="3" x2="6" y2="15"
style="#stroke:000000;stroke-width:1"/>

        </g>
10        <g id="tickMajor" display="none">

                <line x1="0" y1="7" x2="0" y2="17"
style="stroke:#B3B3B3"/>

                <text id="tickLabelMajor" x="0" y="12" xml:space="preserve"
style="font-face:Arial;font-size:8;fill:#B3B3B3;stroke:1;text-anchor:middle"></text>
15        </g>

        <g id="tickMinor" display="none">

                <line x1="0" y1="7" x2="0" y2="11"
style="stroke:#B3B3B3"/>

                </g>
20        </g>

        <g id="mask" >

                <rect id="skinSliderMask" x="-4" y="0" width="208" height="18"
fill="none" stroke="none" pointer-events="fill"/>
25        </g>

</g>

```

</svg>

# skinSpin\_Composite.svg

<?xml version="1.0"?>

5 <svg xmlns:dsvg="http://corel.org/dsvg">

<g id="skinSpin">

< u s e i d = " T e x t B o x S k i n "  
xlink:href="skinTextBox\_Spin\_Short.svg#skinTextbox" x="0" y="0"/>

10 < u s e i d = " B u t t o n S k i n U p "  
xlink:href="skinSpin\_Composite.svg#skinSpinButtonUp" x="100" y="-1"/>

< u s e i d = " B u t t o n S k i n D o w n "  
xlink:href="skinSpin\_Composite.svg#skinSpinButtonDown" x="100" y="8"/>

<g id="label">

<text x="0" y="-6" style="font-size:12;fill:black">

15 <dsvg:constraint preserveAspectRatio="vertical"  
hAlign="middle"/>

</text>

</g>

</g>

20

<g id="skinSpinButtonUp">

<g id="up" display="all">

<rect x="0" y="0" width="18" height="9"  
style="fill:rgb(230,230,230);stroke:none"/>

25 <polyline points="0,9 0,0 18,0" style="stroke-  
width:1;stroke:white;fill:none"/>

```

        <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:black;fill:none"/>

        <polygon points="3,6 9,3 15,6" style="stroke-
width:1;stroke:black;fill:black"/>
5      </g>

      <g id="down" display="none">

        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>

        <polyline points="0,9 0,0 18,0" style="stroke-
10 width:1;stroke:black;fill:none"/>

        <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:white;fill:none"/>

        <polygon points="3,6 9,3 15,6" style="stroke-
width:1;stroke:#199DBF;fill:#199DBF"/>
15      </g>

      <g id="hover" display="none">

        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>

        <polyline points="0,9 0,0 18,0" style="stroke-
20 width:2;stroke:white;fill:none"/>

        <polyline points="0,9 18,9 18,0" style="stroke-
width:2;stroke:black;fill:none"/>

        <polygon points="3,6 9,3 15,6" style="stroke-
width:1;stroke:#199DBF;fill:#199DBF"/>
25      </g>

      <gid="disabled" style="opacity:0.5" transform="scale(0.5)" display="none">

```

```

        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>

        <polyline points="0,9 0,0 18,0" style="stroke-
width:1;stroke:white;fill:none"/>
5        <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:black;fill:none"/>

        <polygon points="3,6 9,3 15,6" style="stroke-
width:1;stroke:black;fill:black"/>

        </g>
10    </g>

    <g id="skinSpinButtonDown">
        <g id="up" display="all">

            <rect x="0" y="0" width="18" height="9"
15    style="fill:rgb(230,230,230);stroke:none"/>

            <polyline points="0,9 0,0 18,0" style="stroke-
width:1;stroke:white;fill:none"/>

            <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:black;fill:none"/>
20        <polygon points="3,3 9,6 15,3" style="stroke-
width:1;stroke:black;fill:black"/>

        </g>

        <g id="down" display="none">

            <rect x="0" y="0" width="18" height="9"
25    style="fill:rgb(230,230,230);stroke:none"/>

            <polyline points="0,9 0,0 18,0" style="stroke-
width:1;stroke:black;fill:none"/>

```

```

        <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:white;fill:none"/>

        <polygon points="3,3 9,6 15,3" style="stroke-
width:1;stroke:#199DBF;fill:#199DBF"/>
5      </g>

      <g id="hover" display="none">

        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>

        <polyline points="0,9 0,0 18,0" style="stroke-
10 width:2;stroke:white;fill:none"/>

        <polyline points="0,9 18,9 18,0" style="stroke-
width:2;stroke:black;fill:none"/>

        <polygon points="3,3 9,6 15,3" style="stroke-
width:1;stroke:#199DBF;fill:#199DBF"/>
15      </g>

      <g id="disabled" style="opacity:0.5" display="none">

        <rect x="0" y="0" width="18" height="9"
style="fill:rgb(230,230,230);stroke:none"/>

        <polyline points="0,9 0,0 18,0" style="stroke-
20 width:1;stroke:white;fill:none"/>

        <polyline points="0,9 18,9 18,0" style="stroke-
width:1;stroke:black;fill:none"/>

        <polygon points="3,3 9,6 15,3" style="stroke-
width:1;stroke:black;fill:black"/>
25      </g>

    </g>

</svg>

```



**skinTextBox\_Combobox.svg**

```

<?xml version="1.0"?>

<svg xmlns:dsvg="http://corel.org/dsvg">

  <g id="skinTextbox">
5      <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
        miterlimit:4;" display="all">
            <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
        </g>
10      <g id="hover" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
            <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
        </g>
15      <g id="down" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
            <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
            <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
20 style="stroke:black">
            <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>
            </line>
        </g>
25      <g id="disabled" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">

```

```

                    <rect x="0" y="0" width="200" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
                </g>
                <clipPath id="skinTextbox_textView">
5          <rect id="skinTextbox_textArea" x="3" y="3" width="205"
height="12"/>
                </clipPath>

                <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
10  xml:space="preserve" x="3" y="15" style="font-face:Arial;font-
size:12;fill:black;stroke:1"></text>

                </g>

```

```

</svg>
15

```

#### **skinTextBox\_Combobox\_Short.svg**

```

<?xml version="1.0"?>
<svg xmlns:dsvg="http://corel.org/dsvg">
20  <g id="skinTextbox">
        <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;" display="all">
                <rect x="0" y="0" width="133" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
25  </g>

```

```

    <g id="hover" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
        <rect x="0" y="0" width="133" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
5        </g>
        <g id="down" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
            <rect x="0" y="0" width="133" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
10        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black">
            <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>
            </line>
15        </g>
        <g id="disabled" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">
            <rect x="0" y="0" width="133" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
20        </g>
        <clipPath id="skinTextbox_textView">
            <rect id="skinTextbox_textArea" x="3" y="3" width="138"
height="12"/>
        </clipPath>
25

```

```

        <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="3" y="15" style="font-face:Arial;font-size:12;fill:black;stroke:1"></text>

    </g>

```

```

5 </svg>

```

### skinTextBox\_Default.svg

```

<?xml version="1.0"?>

10 <svg xmlns:dsvg="http://corel.org/dsvg">
    <g id="skinTextbox">
        <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;stroke-width:1" display="all">
            <rect style="fill-rule:evenodd;clip-
15 rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-miterlimit:2.6131;" x="3" y="3"
width="228" height="18"/>
            <rect style="fill:white;stroke:grey" x="0" y="0" width="228"
height="18"/>
            <dsvg:constraint propertyName="stroke-width"
20 scaleImmunity="true"/>
        </g>
        <g id="hover" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;stroke-width:1;" display="none">
            <rect style="fill-rule:evenodd;clip-
25 rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-miterlimit:2.6131;" x="3" y="3"
width="228" height="18"/>

```

```

    <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
height="18"/>

    <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
5    </g>

    <g id="down" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;stroke-width:1;" display="none">

        <rect style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-miterlimit:2.6131;" x="3" y="3"
10 width="228" height="18"/>

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
height="18"/>

        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black">

15        <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

        <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="top"/>

        </line>

20    <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>

    </g>

    <g id="disabled" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;stroke-width:1;"
25 display="none">

        <rect style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-miterlimit:2.6131;" x="3" y="3"
width="228" height="18"/>

```

```

        <rect style="fill:gray;stroke:#000000" x="0" y="0" width="228"
height="18"/>

        <dsvg:constraint propertyName="stroke-width"
scaleImmunity="true"/>
5         </g>

        <g id="label" style="font-family:'Tahoma';stroke:none;font-size:12">
            <text x="0" y="-6" fill="black"> </text>
            <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="bottom"/>
10        </g>

        <clipPath id="skinTextbox_textView">
            <rect id="skinTextbox_textArea" x="3" y="3" width="222"
height="12"/>
            </clipPath>
15
            <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="3" y="15" style="font-face:Arial;font-
size:12;fill:black;stroke:1"><dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="top"/></text>
20        </g>

        <g id="skinTextboxMultiLine">
            <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;" display="all">
                <rect style="fill-rule:evenodd;clip-
25 rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="3" y="3" width="228" height="54"/>

```

```

        <rect style="fill:white;stroke:grey" x="0" y="0" width="228"
height="54"/>

    </g>

    <g id="hover" style="fill-rule:nonzero;clip-
5 rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

        <rect style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="3" y="3" width="228" height="54"/>

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
10 height="54"/>

    </g>

    <g id="down" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

        <rect style="fill-rule:evenodd;clip-
15 rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="3" y="3" width="228" height="54"/>

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
height="54"/>

        <line id="skinTextboxMultiLine_cursor" x1="3" y1="2" x2="3"
20 y2="16" style="stroke:black">

            <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

            <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="top"/>

25 </line>

    </g>

```

```

    <g id="disabled" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">

        <rect style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
5 x="3" y="3" width="228" height="54"/>

        <rect style="fill:gray;stroke:#000000" x="0" y="0" width="228"
height="54"/>

    </g>

    <g id="label" style="font-family:'Tahoma';stroke:none;font-size:12">
10 <text x="0" y="-6" fill="black"> </text>

        <dsvg:constraint scaleImmunity="true" hAlign="left"
vAlign="bottom"/>

    </g>

    <clipPath id="skinTextboxMultiLine_textView">
15 <rect id="skinTextboxMultiLine_textArea" fill="green" x="3" y="3"
width="222" height="48"/>

    </clipPath>

    <text clip-path="url(#skinTextboxMultiLine_textView)"
id="skinTextboxMultiLine_text" xml:space="preserve" x="3" y="15" style="font-
20 face:Arial;font-size:12;fill:black;stroke:none"><dsvg:constraint scaleImmunity="true"
hAlign="left" vAlign="top"/></text>

    </g>

</svg>

25
skinTextBox_Listbox.svg

<?xml version="1.0"?>

```



```

<svg xmlns:dsvg="http://corel.org/dsvg">

  <g id="skinTextbox">

    <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
      miterlimit:4;" display="all">
5      <rect x="0" y="0" width="200" height="17"
        style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
      </g>
      <g id="hover" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
10      <rect x="0" y="0" width="200" height="17"
        style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
      </g>
      <g id="down" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">
15      <rect x="0" y="0" width="200" height="17"
        style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
          style="stroke:black">
          <animate attributeType="CSS" attributeName="opacity"
20 values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>
        </line>
      </g>
      <g id="disabled" style="fill-rule:nonzero;clip-
        rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">
25      <rect x="0" y="0" width="200" height="17"
        style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>
      </g>

```

```

    <clipPath id="skinTextbox_textView">
        <rect id="skinTextbox_textArea" x="3" y="3" width="205"
height="12"/>
    </clipPath>

```

5

```

        <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="3" y="15" style="font-face:Arial;font-
size:12;fill:black;stroke:1"></text>
    </g>

```

10

```

</svg>

```

### skinTextBox\_Spin\_Short.svg

15

```

<?xml version="1.0"?>

```

```

<svg xmlns:dsvg="http://corel.org/dsvg">

```

```

    <g id="skinTextbox">

```

```

        <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;" display="all">

```

20

```

            <rect x="0" y="0" width="100" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

```

```

        </g>

```

```

        <g id="hover" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

```

25

```

            <rect x="0" y="0" width="100" height="17"
style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

```

```

    </g>

    <g id="down" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

        <rect x="0" y="0" width="100" height="17"
5 style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black">

            <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>
10 </line>

    </g>

    <g id="disabled" style="fill-rule:nonzero;clip-
rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">

        <rect x="0" y="0" width="133" height="17"
15 style="fill:#FFFFFF;stroke:#A3A3A3;stroke-width:1"/>

    </g>

    <clipPath id="skinTextbox_textView">

        <rect id="skinTextbox_textArea" x="3" y="3" width="108"
height="12"/>
20 </clipPath>

        <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="3" y="15" style="font-face:Arial;font-
size:12;fill:black;stroke:1"></text>
25 </g>

</svg>

```

**skinTextBox\_vp.svg**

```

<?xml version="1.0"?>

<svg>
5      <g id="skinTextbox">

          <g id="up" style="fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
            miterlimit:4;" display="all">

              <rect style="fill-rule:evenodd;clip-
                rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
10      x="3" y="3" width="228" height="18"/>

              <rect style="fill:white;stroke:grey" x="0" y="0" width="228"
                height="18"/>

              </g>

              <g id="hover" style="fill-rule:nonzero;clip-
15      rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

                  <rect style="fill-rule:evenodd;clip-
                    rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
                    x="3" y="3" width="228" height="18"/>

                  <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
20      height="18"/>

                  </g>

                  <g id="down" style="fill-rule:nonzero;clip-
                    rule:nonzero;stroke:#000000;stroke-miterlimit:4;" display="none">

                      <rect style="fill-rule:evenodd;clip-
25      rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
                      x="3" y="3" width="228" height="18"/>

```

```

        <rect style="fill:white;stroke:#000000" x="0" y="0" width="228"
height="18"/>

        <line id="skinTextbox_cursor" x1="3" y1="2" x2="3" y2="16"
style="stroke:black"/>
5          <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

        </line>

        </g>

        <g id="disabled" style="fill-rule:nonzero;clip-
10 rule:nonzero;stroke:#000000;stroke-miterlimit:4;opacity:0.5;" display="none">

            <rect style="fill-rule:evenodd;clip-
rule:evenodd;fill:#FFFFFF;stroke:#B3B3B3;stroke-width:1.0001;stroke-miterlimit:2.6131;"
x="3" y="3" width="228" height="18"/>

            <rect style="fill:gray;stroke:#000000" x="0" y="0" width="228"
15 height="18"/>

        </g>

        <g id="label">

            <text x="238" y="14" style="font-size:12;fill:black"> </text>

        </g>
20    <clipPath id="skinTextbox_textView">

        <rect id="skinTextbox_textArea" x="3" y="3" width="222"
height="12"/>

        </clipPath>

25    <text clip-path="url(#skinTextbox_textView)" id="skinTextbox_text"
xml:space="preserve" x="3" y="15" style="font-face:Arial;font-
size:12;fill:black;stroke:1"></text>

```

```

    </g>

    <g id="skinTextboxMultiLine">

        <g id="up" display="all">

            <rect stroke="#EEEEEE" stroke-width="2" fill="white" x="0" y="-5"
5      width="380" height="195"/>

            <line stroke="black" x1="1" y1="-4" x2="380" y2="-4"/>

            <line stroke="black" x1="1" y1="-4" x2="1" y2="191"/>

            <rect stroke="black" fill="white" x="2" y="-3" width="377"
height="192"/>
10      </g>

            <g id="hover" display="none">

                <rect stroke="#EEEEEE" stroke-width="2" fill="white" x="0" y="-5"
width="380" height="195"/>

                <line stroke="black" x1="1" y1="-4" x2="380" y2="-4"/>
15      <line stroke="black" x1="1" y1="-4" x2="1" y2="191"/>

                <rect stroke="black" fill="white" x="2" y="-3" width="377"
height="192"/>
                </g>

            <g id="down" display="none">

                <rect stroke="#EEEEEE" stroke-width="2" fill="white" x="0" y="-5"
20      width="380" height="195"/>

                <line stroke="black" x1="1" y1="-4" x2="380" y2="-4"/>

                <line stroke="black" x1="1" y1="-4" x2="1" y2="191"/>

                <rect stroke="black" fill="white" x="2" y="-3" width="377"
height="192"/>
25      <line id="skinTextboxMultiLine_cursor" x1="3" y1="2" x2="3"
y2="16" style="stroke:black">

```

```

        <animate attributeType="CSS" attributeName="opacity"
values="0;0;0;0;0;0;1;1;1;1;1;1" dur="1s" repeatCount="indefinite"/>

        </line>

        </g>
5         <g id="disabled" display="none">

        <rect stroke="gray" fill="white" x="0" y="-5" width="380" height="195"/>

        </g>

        <clipPath id="skinTextboxMultiLine_textView">

        <rect id="skinTextboxMultiLine_textArea" fill="green" x="3" y="3"
10 width="373" height="185"/>

        </clipPath>

        <text clip-path="url(#skinTextboxMultiLine_textView)"
id="skinTextboxMultiLine_text" xml:space="preserve" x="7" y="15" style="font-
face:Arial;font-size:12;fill:black;stroke:none"></text>
15         </g>

</svg>

skinTextScroll_Default.svg
20 <?xml version="1.0"?>

<svg>

        <g id="skinTextboxScroll">

                < u s e      i d = " T e x t B o x S k i n "
xlink:href="skinTextBox_Default.svg#skinTextboxMultiLine" x="0" y="0"/>
25                < u s e      i d = " V e r t S c r o l l S k i n "
xlink:href="skinScrollBar_Default.svg#skinScrollBarSmall" x="0" y="0"/>

```

```

        < u s e       i d = " H o r i z S c r o l l S k i n "
xlink:href="skinScrollBar_Default.svg#skinScrollBarSmall" x="0" y="0"/>

```

```

    </g>

```

```

    <g id="label" style="font-family:'Tahoma';stroke:none;font-size:12">

```

```

5        <text x="0" y="-6" fill="black"> </text>

```

```

        <dsvg:constraint scaleImmunity="true" hAlign="left" vAlign="bottom"/>

```

```

    </g>

```

```

</svg>

```

# 10 skinTooltip.svg

```

<?xml version="1.0"?>

```

```

<svg>

```

```

    <g id="skinTooltip_annotation">

```

```

        <path style="fill:#FFFF00;fill-opacity:0.7;stroke:#1F1A17;stroke-width:3"

```

```

15        d="M0 96 198 0 -56 -96 106 96 202 0 0 167 -352 0 0 -167z"/>

```

```

        <text x="10" y="150" style="font-size:16;fill:black">tip</text>

```

```

    </g>

```

```

    <g id="skinTooltip_traditional" pointer-events="none">

```

```

20        <rect x="0" y="0" width="150" height="15"

```

```

        style="fill:rgb(255,255,225);fill-opacity:0.9" stroke="black" stroke-width="1"/>

```

```

        <text x="5" y="12" style="font-size:12;fill:black">tip</text>

```

```

    </g>

```

```

25    <g id="skinTooltip_special" pointer-events="none">

```



```
<rect x="0" y="0" width="150" height="15"
```

```
style="fill:rgb(130,130,130);fill-opacity:0.5" stroke="black" stroke-width="1"/>
```

```
<text x="5" y="12" style="font-size:12;fill:black">tip</text>
```

```
</g>
```

```
5 </svg>
```

## WHAT IS CLAIMED IS:

1. A system for controlling user interface features of a web application, the system comprising:

- 5           a collection of user interface control elements, each control element comprising:
- a name following a predetermined naming convention;
- common attributes for defining graphical features of the control element and for associating the control element with the internal state of the core control element;
- other attributes for defining attributes that affect the intrinsic behavior of the
- 10   control; and
- a skin template reference attribute for referencing a skin template;
- a collection of skin templates comprising extensible markup language based markup contained as children of a container element;
- a collection of control element scripts for performing actions associated with the
- 15   control elements, each script associated with a control element; and
- an initialization function for directing the processing one or more control elements in a document object model.

20   2. The system as claimed in claim 1, wherein the predetermined naming convention comprises having a constant prefix to the name of the element.

3. The system as claimed in claim 1, wherein the skin template reference attribute comprises a reference to the location of a skin template file.

25   4. The system as claimed in claim 1, wherein the control element is associated with an extensible markup language based element.

5. The system as claimed in claim 4, wherein the control element is a parent of an extensible markup language based element.

6. The system as claimed in claim 4, wherein the control element is a child of an extensible  
5 markup language based element.

7. The system as claimed in claim 1, wherein the initialization function contains instructions for traversing each node in a document object model and for searching and calling functions associated with control elements having names following the predetermined naming  
10 convention.

8. The system as claimed in claim 1, further comprising:

a collection of control attributes for adding to existing regular extensible markup language based elements in a document object model, the control attributes following the  
15 predetermined naming convention; and

a collection of control attribute scripts for performing actions associated with the collection of control attributes, each script associated with a control attribute.

9. The system as claimed in claim 8, wherein the initialization function contains instructions  
20 for traversing each node in the document object model and for searching and calling functions associated with control elements and control attributes having names following the predetermined naming convention.

10. The system as claimed in claim 8, wherein the collection of control attribute scripts is  
25 stored in a memory location of a computer system.

11. The system as claimed in claim 1, wherein the collection of scripts is stored in a memory location of a computer system.

12. The system as claimed in claim 1, wherein the collection of skin templates is stored in  
5 a memory location of a computer system.

13. The system as claimed in claim 1, wherein the collection of control elements comprises a markup language.

10 14. The system as claimed in claim 1, wherein the common attributes comprise state attributes for specifying the identification of a <state> child element of the control element.

15. The system as claimed in claim 1, wherein the common attributes comprise one or more of:

15 an identification attribute for referencing the control element;

a label attribute for associating text control;

an x attribute for specifying the x-coordinate of the left edge of the control element;

a y attribute for specifying the y-coordinate of the top edge of the control element;

a width attribute for specifies the width of the control element;

20 a height attribute for specifies the height of the control element;

a preserve aspect ratio attribute for preserving the aspect ratio of the control element when either the width attribute or height attribute is known;

a labelX attribute for specifying the x-coordinate of the left edge of the label, relative to the 'y' attribute;

25 a labelY attribute for specifying the y-coordinate of the bottom edge of the label, relative to the 'x' attribute;

a disabled attribute for specifying whether the control element is disabled and cannot be used;

5 a state hover attribute for specifying the identification of a <state> child element of the control element, the state hover attribute used to override the appearance of a hover state as defined in a skin of the control element;

a state focus attribute for specifying the identification of a <state> child element of the control element, the state focus attribute used to override the appearance of a focus state as defined in a skin of the control element;

10 a state up attribute for specifying the identification of a <state> child element of the control element, the state up attribute used to override the appearance of an up state as defined in a skin of the control element;

a state down attribute for specifying the identification of a <state> child element of the control element, the state down attribute used to override the appearance of a down state as defined in a skin of the control element;

15 a state hit attribute for specifying the identification of a <state> child element of the control element, the state hit attribute used to override the appearance of a hit state as defined in a skin of the control element;

20 a state disabled up attribute for specifying the identification of a <state> child element of the control element, the state disabled up attribute used to override the appearance of a disabled up state as defined in the skin of the control element; and

a state disabled down attribute for specifying the identification of a <state> child element of the control element, the state disabled down attribute used to override the appearance of a disabled down state as defined in a skin of the control element

25 16. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:button control element for defining a control that is clicked to trigger an action, the dsvg:button control element comprising:

a namespace following the predetermined naming convention;

the common attributes;

other attributes comprising:

a toggle attribute for specifying whether the button is a toggle or a sticky button;

5 a group attribute for specifying the name of a group to which the button control element belongs; and

a checked attribute for specifying whether the button control element is down/checked or up/unchecked;

10 a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index; and

a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

17. The system as claimed in claim 15, wherein the set of control elements comprises a  
15 dsvg:comboBox control element for defining a control that is clicked to trigger an action, the dsvg:comboBox control element comprising:

a namespace following the predetermined naming convention;

the common attributes;

other attributes comprising:

20 a dropdown attribute for specifying whether the comboBox control element has a dropdown list;

an editable attribute for specifying whether the comboBox control element is editable;

25 a value attribute for specifying the value of the label attribute of currently selected item;

a name attribute for specifying the value of a name attribute of a currently selected item; and

a selected identification attribute for specifying the value of the identification attribute of a currently selected item;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index; and

5 a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

18. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:listBox control element for defining a control that is clicked to trigger an action, the  
10 dsvg:listBox control element comprising:

a namespace following the predetermined naming convention;

the common attributes;

other attributes comprising:

15 a multi select attribute for specifying whether more than one item can be selected;

an editable attribute for specifying whether the listBox control element is editable;

a value attribute for specifying the value of the label attribute of currently selected item;

20 a name attribute for specifying the value of a name attribute of a currently selected item; and

a selected identification attribute for specifying the value of the identification attribute of a currently selected item;

25 a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

19. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:ListView control element for defining a control that is clicked to trigger an action, the dsvg:ListView control element comprising:

- a namespace following the predetermined naming convention;
- 5 the common attributes;
- other attributes comprising:
  - a multi select attribute for specifying whether more than one item can be selected;
  - an editable attribute for specifying whether the comboBox control element is
  - 10 editable;
  - a display attribute for specifying a semicolon-delimited list of the names of the attributes in all of the <item> children of a control element;
  - an anything attribute for specifying an attribute name in which to store data;
  - and
  - 15 a selected identification attribute for specifying the value of the identification attribute of a currently selected item;
  - a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;
  - a customizable skin template comprising scalable vector graphics markup contained
  - 20 as children of a container element.

20. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:ContextMenu control element for defining a control that is clicked to trigger an action, the dsvg:ContextMenu control element comprising:

- 25 a namespace following the predetermined naming convention;
- the common attributes;
- other attributes comprising:



an event source attribute for specifying the identification of an element that triggered the contextMenu control element to appear;

a value attribute for specifying the value of the label attribute of currently selected item;

5 a name attribute for specifying the value of a name attribute of a currently selected item; and

a selected identification attribute for specifying the value of the identification attribute of a currently selected item;

10 a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

21. The system as claimed in claim 15, wherein the set of control elements comprises a  
15 dsvg:item control element for defining a control that is clicked to trigger an action, the dsvg:item control element comprising:

a namespace following the predetermined naming convention;

the common attributes;

other attributes comprising:

20 an access key attribute for specifying a shortcut key which, when pressed, selects this item; and

an anything attribute for specifying an attribute name in which to store data;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

25 a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

22. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:textbox control element for defining a control that is clicked to trigger an action, the dsvg:textbox control element comprising:

a namespace following the predetermined naming convention;

5 the common attributes;

other attributes comprising:

a value attribute for specifies default text within the textbox control element;

a num lines attribute for specifying a number of lines allowed in the textbox control element;

10 a max length attribute for specifying a maximum number of characters allowed in the textbox control element;

a wrap attribute for specifying whether to auto-wrap text;

a read only attribute for specifying if the textbox control element is non-editable;

15 a secret attribute for specifies whether text is secret;

a data type attribute for specifying a type of data that allowed to be entered;

a mask attribute for specifies a pattern that allows extra characters to be inserted into data as it is entered and for only allowing specific characters in specific locations; and

20 a case attribute for specifying the case of data entered into the textbox control element;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

25 a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

23. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:slider control element for defining a control that is clicked to trigger an action, the dsvg:slider control element comprising:

a namespace following the predetermined naming convention;

5 the common attributes;

other attributes comprising:

a min attribute for specifying a minimum value of the slider control element;

a max attribute for specifying a maximum value of the slider control element;

a min position attribute for specifies a minimum allowed value of a thumb;

10 a max position attribute for specifies a maximum allowed value of a thumb;

a value attribute for specifying an initial value of the slider control element;

an increment attribute for specifying allowed values that the slider control element can create;

a page increment attribute for specifying an amount that a thumb moves;

15 an orientation attribute for specifying a rotation angle of the slider control attribute;

a ticks major attribute for specifying an interval at which major tick marks are displayed; and

20 a ticks minor attribute for specifying an interval at which minor tick marks are displayed;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

25

24. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:scrollbar control element for defining a control that is clicked to trigger an action, the dsvg:scrollbar control element comprising:

a namespace following the predetermined naming convention;

5 the common attributes;

other attributes comprising a bars attribute for specifies the appearance of a horizontal scrollbar or a vertical scrollbar;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

10 a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

25. The system as claimed in claim 15, wherein the set of control elements comprises a dsvg:spin control element for defining a control that is clicked to trigger an action, the dsvg:spin control element comprising:

15

a namespace following the predetermined naming convention;

the common attributes;

other attributes comprising:

a min attribute for specifying a minimum value of the spin control element;

20

a max attribute for specifying a maximum value of the spin control element;

a value attribute for specifying an initial value of the spin control element; and

an increment attribute for specifying allowed values that the spin control element can create;

a skin template reference attribute for specifying the location of a control element skin template, the skin template reference settable to a uniform resource index;

25

a customizable skin template comprising scalable vector graphics markup contained as children of a container element.

26. A system for controlling user interface features of a web application, the system comprising:

a collection of control element scripts for performing actions associated with the control elements, each script associated with a control element; and

5 an initialization function for directing the processing of one or more control elements in a document object model.

27. The system as claimed in claim 26, further comprising a collection of skin templates comprising extensible markup language based markup contained as children of a container  
10 element.

28. A method of controlling user interface features of a web application, the method comprising the steps of:

searching for a designated user interface control element in a document object model;

15 and

calling a script associated with the designated control element.

29. The method as claimed in claim 28, wherein the step of searching includes the steps of:

traversing each node in the document object model; and

20 determining whether an element has a name which follows a designated naming convention.

30. The method as claimed in claim 29, wherein the designated naming convention comprises appending a prefix to the name of the designated element.

25

31. The method as claimed in claim 28, wherein the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated element;  
passing an object associated with the designated element as a parameter of the  
generated function;  
retrieving the attributes of the object; and  
5 performing a function stored in memory having the generated function name.

32. The method as claimed in claim 31, wherein the step of dynamically generating includes the steps of:

determining if the name of the designated element contains a designated prefix;  
10 generating a function name comprising of the name of the designated element;  
assigning an object associated with the designated element as the parameter of the  
function; and  
assigning predetermined instructions of the designated element as steps for the  
function to perform.

15

33. The method as claimed in claim 28, wherein the step of calling a script includes the steps of:

determining which script in a collection of scripts is associated with the designated  
element; and  
20 calling the script.

34. The method as claimed in claim 28, further comprising the steps of:

searching for a designated attribute in an element in a document object model; and  
calling a script associated with the designated attribute.

25

35. The method as claimed in claim 34, wherein the step of searching for a designated attribute comprises the steps of:

searching attributes of an element in a document object model;

5 determining whether an element attribute has a name which follows a designated naming convention.

36. The method as claimed in claim 35, wherein the naming convention comprises appending a prefix to the name of the designated attribute.

10 37. The method as claimed in claim 34, wherein the step of calling a script includes the steps of:

determining if the name of the designated attribute contains a designated prefix;

generating a function name comprising of the name of the designated attribute;

15 assigning an object associated with the designated attribute as the parameter of the function name ; and

assigning predetermined instructions of the designated attribute as steps for a function having the function name to perform.

20 38. The method as claimed in claim 34, wherein the step of calling a script includes the steps of:

dynamically generating a function name associated with the designated attribute;

passing an object associated with the designated attribute as a parameter of the generated function name;

receiving the attributes of the object; and

25 performing a function stored in memory having the generated function name.

39. The method as claimed in claim 38, wherein the step of dynamically generating comprises the steps of:

determining if the name of the designated attribute contains a designated prefix;

generating a function name comprising of the name of the designated attribute;

5 assigning an object associated with the designated attribute as the parameter of the function; and

assigning predetermined instructions of the designated attribute as steps for the function to perform.

10 40. The method as claimed in claim 34, wherein the step of calling a script includes the steps of:

determining which script in a collection of scripts is associated with the designated attribute; and

calling the script.

15

41. A method of controlling user interface features of a web application, the method comprising the steps of:

adding a behavior element as a child of a user interface control element;

receiving an event which is equal to an event attribute setting in the behavior element;

20 and

calling a script associated with the behavior element.

42. A method of creating a customizable user interface control element having expected behaviours, the method comprising the steps of:

25 categorising user interface controls into fundamental core controls;

determining variations of a core controls;



determining common attributes of the core control;

determining fundamental states for the core control;

determining how to allow for absolute positioning of objects the core control;

5 determining how to allow for absolute customization of appearance of the core control;

assigning a reference link to the core control;

determining templates for skins to allow for the absolute customization of appearance of the core control;

determining how to associate behaviors to the core control; and

10 creating a core control element.

43. The method as claimed in claim 42, wherein a plurality of user interface control elements are created.

15 44. The method as claimed in claim 42, wherein the step of categorising includes the steps of:

organising low level controls into groups of similar controls; and

designating core control names to the groups.

20 45. The method as claimed in claim 42, wherein the step of determining includes the steps of:

analysing variations of the core controls; and

compiling a list of attributes needed to perform the variations.

25 46. The method as claimed in claim 42, wherein the step of creating includes the steps of:

assigning a name to the core control element pursuant to a predetermined naming convention; and

assigning the common attributes to the core control element.

- 5 47. Computer readable media storing the instructions and/or statements for use in the execution in a computer of a method of controlling user interface features of a web application, the method comprising steps of:

searching for a designated user interface control element in a document object model;  
and

- 10 calling a script associated with the designated control element.

48. Electronic signals for use in the execution in a computer of a method of controlling user interface features of a web application, the method comprising steps of:

- 15 searching for a designated user interface control element in a document object model;  
and

calling a script associated with the designated control element.

49. A computer program product for use in the execution in a computer of a method for controlling user interface features of a web application, the computer program product  
20 comprising:

a collection of control element scripts for performing actions associated with the control elements, each script associated with a control element; and

an initialization function for directing the processing of one or more control elements in a document object model.

25

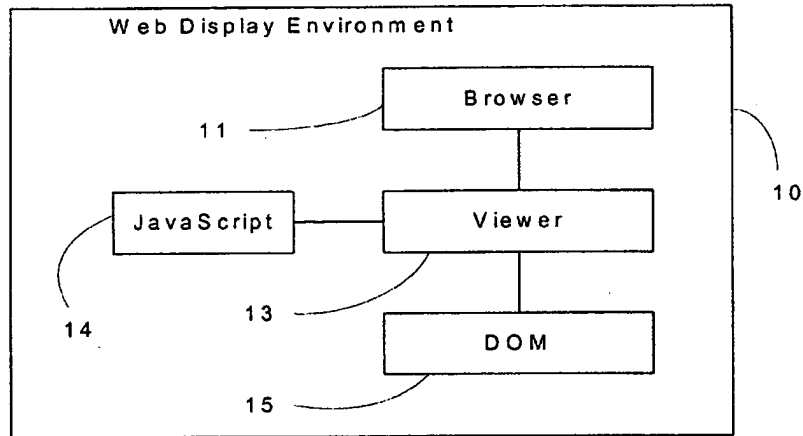


Figure 1  
PRIOR ART

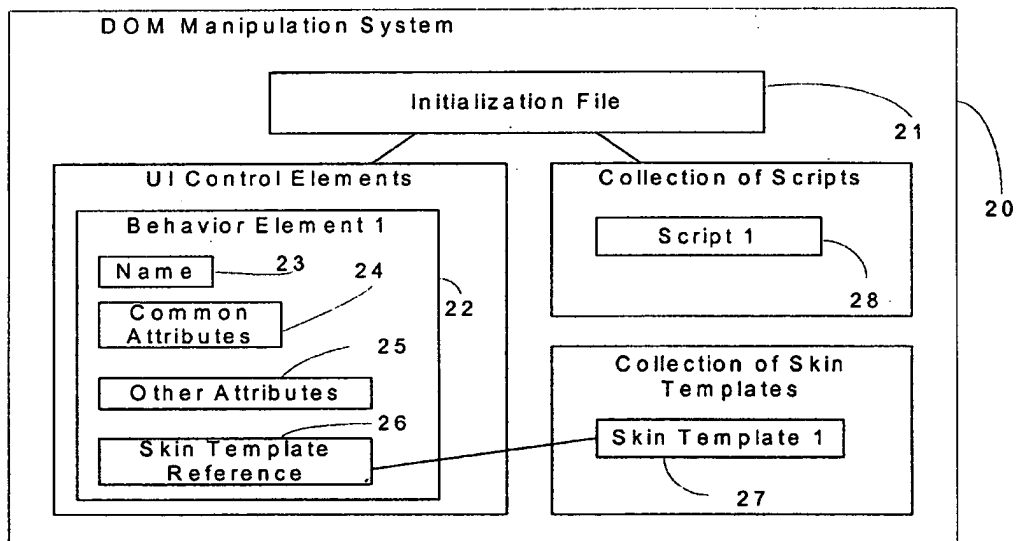


Figure 2

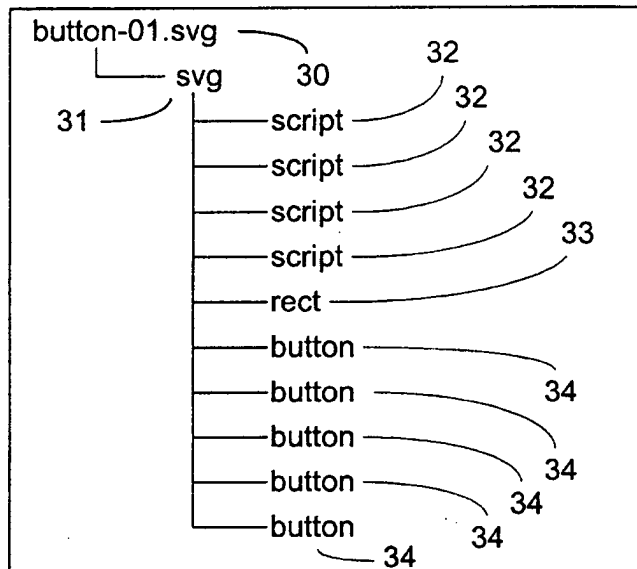


Figure 3

5

10

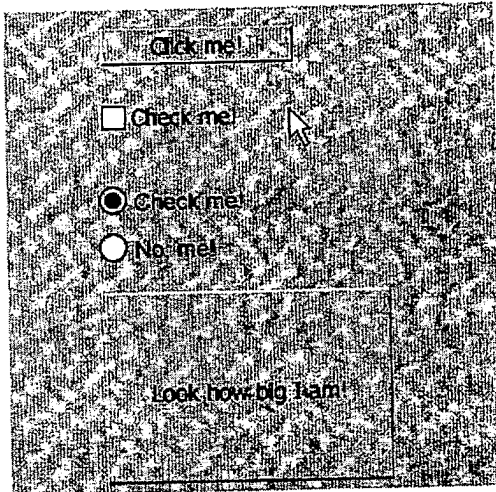


Figure 4A

15

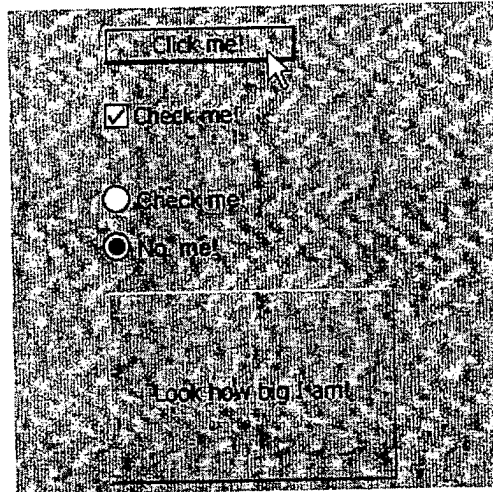


Figure 4B

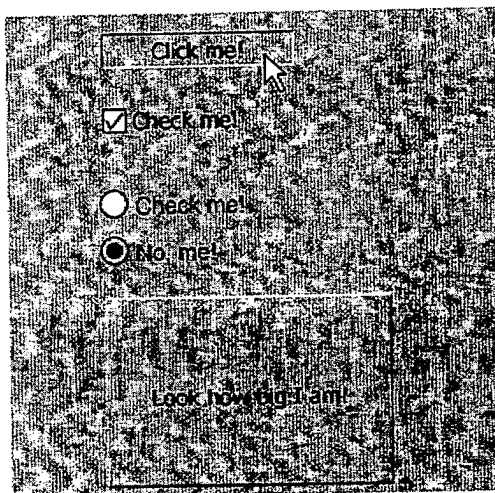


Figure 4C

20

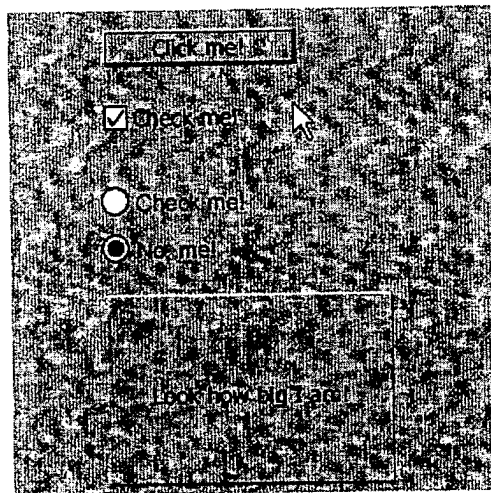


Figure 4D

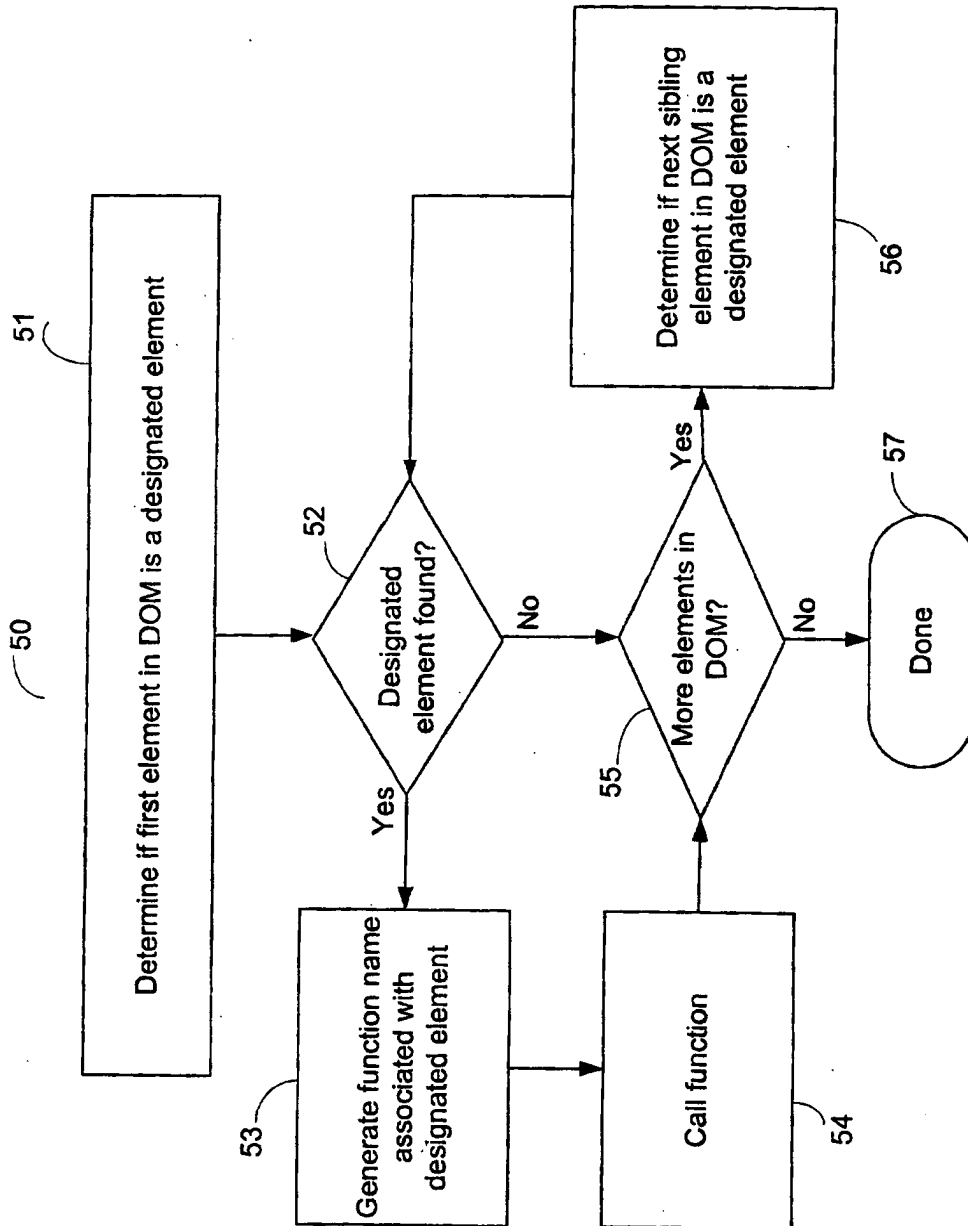


Figure 5

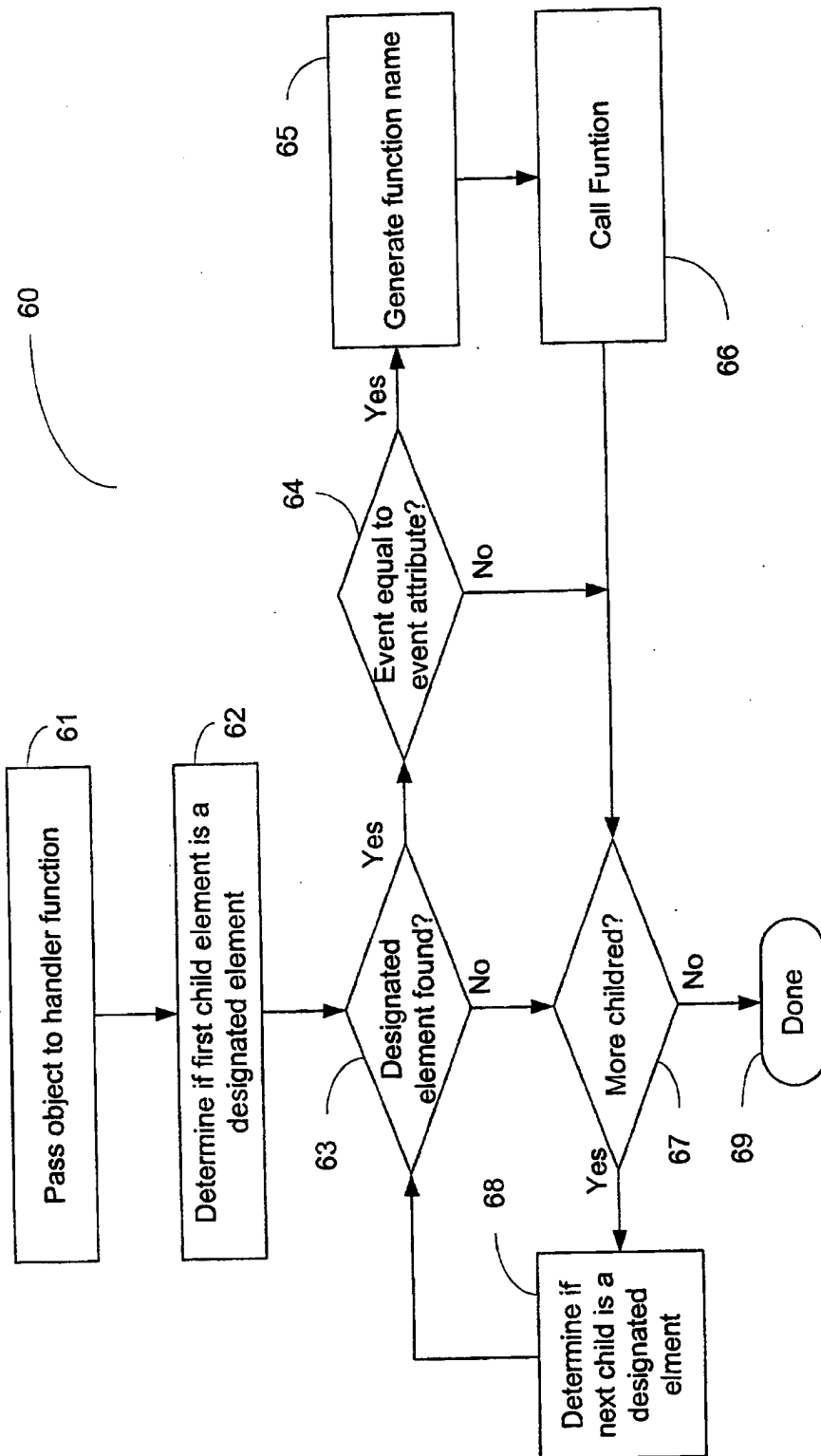
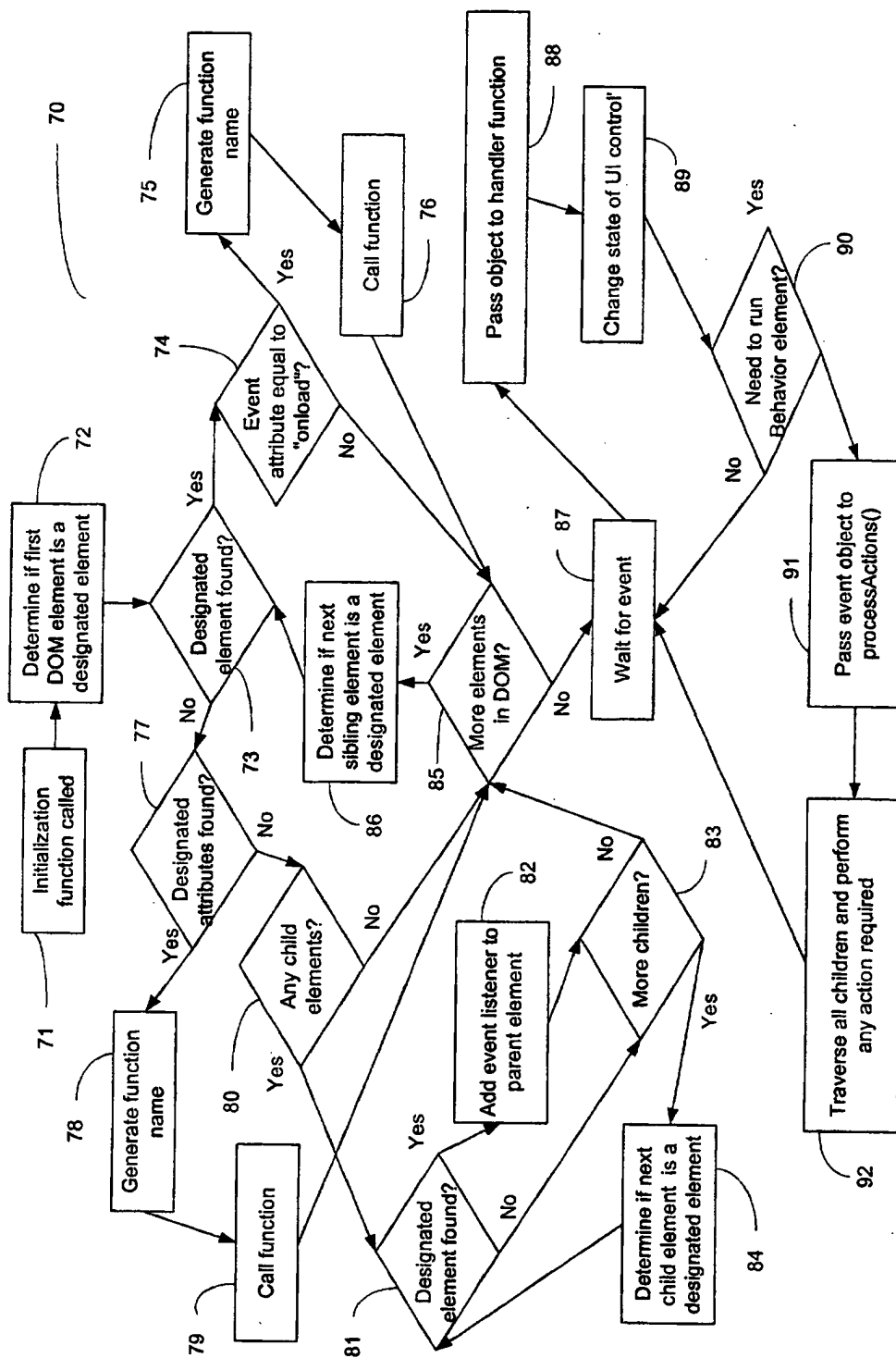


Figure 6



### Figure 7



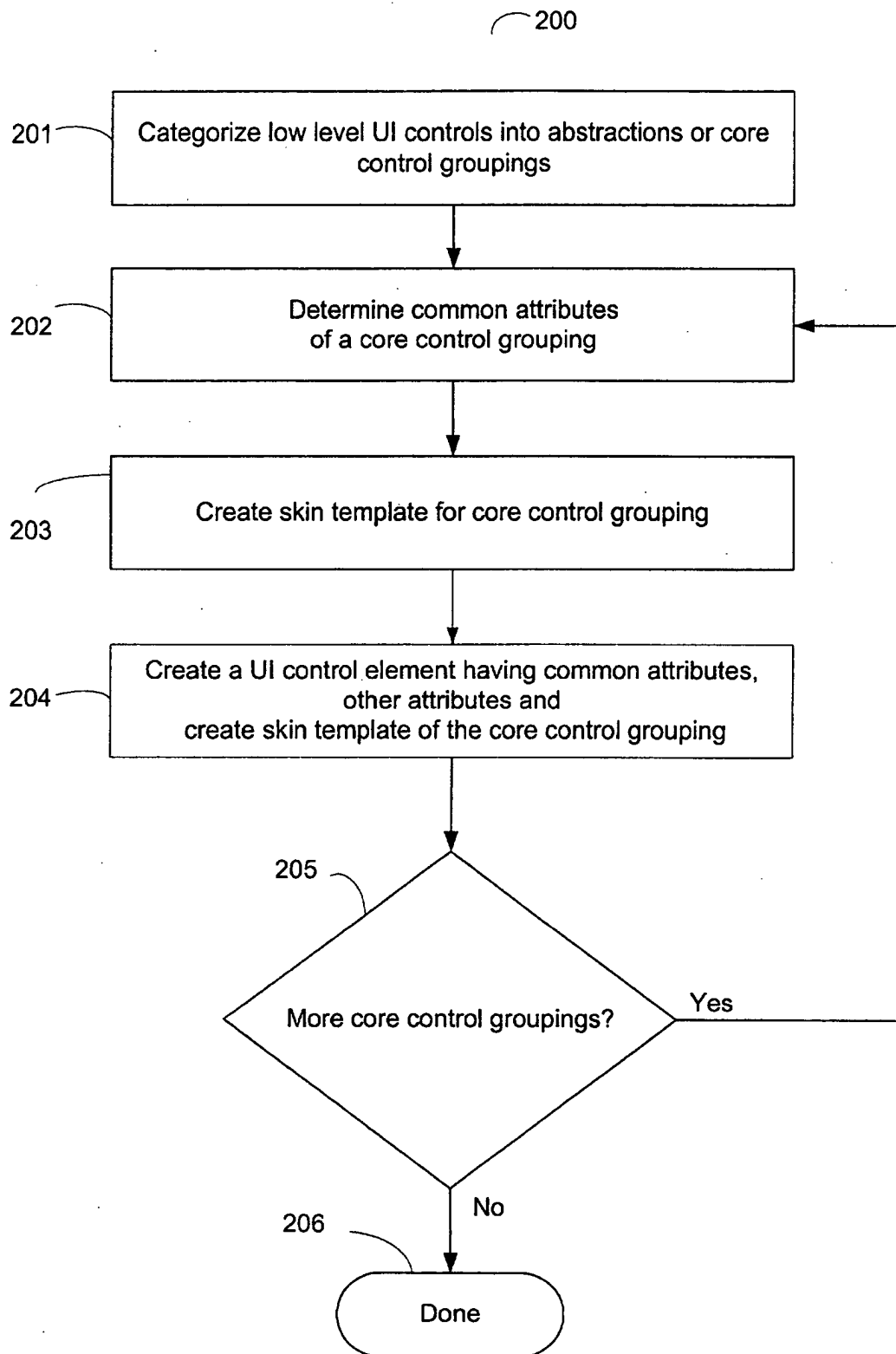


Figure 8

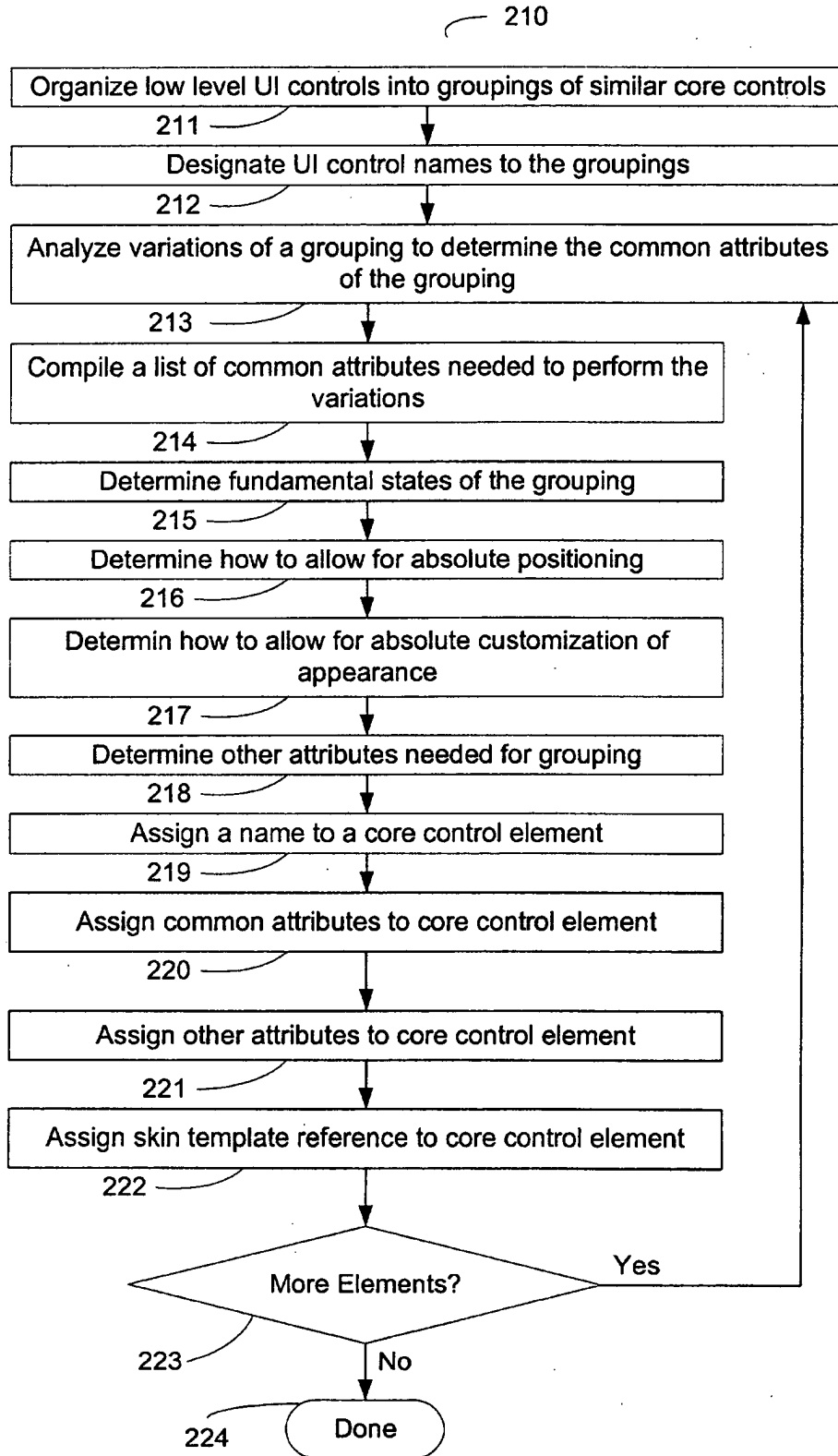


Figure 9